THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue le *Date de défense (29/10/2021)* **par :**

AMANDINE MAYIMA

Endowing the Robot with the Abilities to Control and Evaluate its Contribution to a Human-Robot Joint Action

---

JURY

SILVIA ROSSI	Professeure Associée	Rapporteure
PETER FORD DOMINEY	Directeur de Recherche	Rapporteur
RACHID ALAMI	Directeur de Recherche	Directeur de Thèse
AURÉLIE CLODIC	Ingénieure de Recherche	Directrice de Thèse
SIMON LACROIX	Directeur de Recherche	Membre du Jury
GUY HOFFMAN	Professeur Associé	Membre du Jury
ELISABETH PACHERIE	Directrice de Recherche	Membre du Jury

École doctorale et spécialité :
MITT : Informatique et Télécommunications
Unité de Recherche :
Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)
Directeur(s) de Thèse :
Rachid ALAMI et *Aurélie CLODIC*
Rapporteurs :
Silvia ROSSI et *Peter Ford DOMINEY*

Abstract

Robots will interact more and more with humans in the future and thus will need to be endowed with the pertinent abilities. We are still far from having autonomous robots among humans and able to smoothly collaborate with them but the work of this thesis is a contribution bringing the community a bit closer to this goal.

When humans collaborate to achieve a task together, numerous neurocognitive mechanisms come into play, more than we would have thought at first glance. Some of these mechanisms are also triggered in humans' minds when they interact with robots as they are essential to a successful collaboration. Therefore, it is important for roboticists designing robots that will closely interact with humans to be aware of and take into account the humans mental states and sensorimotor functions involved in controlling and smoothing collaborative task performance. However, this does not imply that robots have to be endowed with the same mechanisms since being able to collaborate with humans does not mean to imitate them. What is key to roboticists is to understand how humans work and to design robots that will adapt.

Consequently, this manuscript starts with an immersion in philosophy and social and cognitive psychology. We develop key elements for collaboration such as joint action, commitment and shared plans. Then, we explore Belief-Desire-Intention (BDI) and cognitive robotic architectures which have inspired us to design our own architecture in which, JAHRVIS — the main contribution of this thesis — endows a robot with the abilities not only to control, but also to evaluate its joint action with a human.

JAHRVIS (Joint-Action based Human-Aware supeRVISor) is what we call a supervision system, *i.e.*, , it embeds the robot high-level decisions, controls its behavior and tries to react to contingencies, always considering the human it is interacting with. It is able to do so by taking into account shared plans, human mental states, its knowledge about the current state of the environment, and human actions. The module to monitor and recognize these actions is model-based and allows to take into account a potentially unreliable perception of the human. JAHRVIS is designed in such a way that it is generic enough to handle various kinds of tasks. It can also manage different kinds of human-robot shared plans as input: shared plans for which actions might not be allocated to an agent at planning time and objects might be referred to with a semantic query, and conditional shared plans which anticipate different possibilities for the human decision/action.

Not only JAHRVIS controls the robot contribution to a collaborative task, it also tries to evaluate if the interaction is going well or not. It is possible thanks to a set of metrics we have built and a method to aggregate them. We claim that having a robot with this ability allows it to enhance and make more pertinent its decision-making processes. The evaluation of the Quality of Interaction (QoI) relies on a model of interaction, considered at three levels: the interaction session level, the tasks level and the actions level. In future work, this granularity will allow the

robot to know precisely on what level it needs to act when a low QoI is assessed. Indeed, for instance, a task can be of poor quality while the session can still be considered as going well.

JAHRVIS has been integrated in a cognitive robotic architecture and effectively deployed to achieve several collaborative and service tasks in a real environment such as a direction-giving task in a Finnish mall and a two-agents task inspired from psychology. These tasks demonstrated the robot's abilities related to perspective-taking, planning, knowledge representation with theory of mind, manipulation, and communication.

Keywords: Human-Robot interaction, Joint Action, Decision making, Quality of Interaction

Résumé

Dans le futur, les robots interagiront chaque jour un peu plus avec les humains et devront donc être dotés des capacités adéquates. Nous sommes encore loin de robots autonomes parmi les humains, capables de collaborer sans problème avec eux: le travail de cette thèse est une contribution qui rapproche un peu plus la communauté de cet objectif.

Lorsque des personnes collaborent pour réaliser une tâche ensemble, de nombreux mécanismes neurocognitifs entrent en jeu, plus qu'il n'y paraît à première vue. Certains de ces mécanismes sont aussi activés quand un humain interagit avec un robot et non plus avec un autre humain, car ils sont essentiels à une collaboration réussie. Il est donc important que les roboticiens qui conçoivent des robots destinés à interagir étroitement avec les humains soient conscients de cela et qu'ainsi ils prennent en compte les états mentaux des humains et les fonctions sensori-motrices impliquées dans le contrôle et la fluidité de l'exécution des tâches collaboratives. Toutefois, cela ne signifie pas que les robots doivent être dotés de ces mêmes mécanismes, car être capable de collaborer avec les humains ne signifie pas les imiter. Ce qui est essentiel pour les roboticiens, c'est de comprendre comment les humains travaillent et de concevoir des robots qui s'adapteront.

Ce manuscrit commence par une immersion dans la philosophie et la psychologie sociale et cognitive. Nous développons les éléments clés de la collaboration tels que l'action conjointe, l'engagement et les plans partagés. Ensuite, nous explorons les modèles "croyance-désir-intention" (Belief-Desire-Intention en anglais) et les architectures robotiques cognitives qui nous ont inspirés pour concevoir notre système de supervision, JAHRVIS - la principale contribution de cette thèse qui dote un robot des capacités de non seulement de contrôler, mais aussi d'évaluer son action conjointe avec un humain.

Joint Action-based Human-aware supeRVISor (JAHRVIS) est ce que nous appelons un système de supervision, c'est-à-dire qu'il prend les décisions haut niveau du robot, contrôle son comportement et tente de réagir aux imprévus, en tenant toujours compte de l'humain avec lequel il interagit. Il peut le faire en se basant sur les plans partagés qu'il génère, sa connaissance des états mentaux de l'humain et de l'état actuel de l'environnement et, les actions de l'humain. Le module de surveillance et de reconnaissance de ces actions est basé sur un modèle et permet au robot de prendre en compte une perception potentiellement peu fiable. JAHRVIS est conçu de manière à être suffisamment générique pour gérer différents types de tâches. Il peut également gérer différents types de plans partagés homme-robot en entrée : des plans partagés pour lesquels les actions peuvent ne pas être attribuées à un agent au moment de la planification et pour lesquels les objets peuvent être référencés par une requête sémantique, et des plans partagés conditionnels qui anticipent différentes possibilités pour la décision/action humaine.

JAHRVIS ne se contente pas de contrôler la contribution du robot à une tâche

collaborative, il essaie également d'évaluer si l'interaction se déroule bien ou non. Cela est possible grâce à un ensemble de métriques et à une méthode pour les agréger que nous avons conçue. Nous affirmons que le fait de doter un robot de cette capacité lui permet d'améliorer et de rendre plus pertinent son processus de prise de décision. L'évaluation de la qualité d'interaction (QoI) repose sur un modèle d'interaction à trois niveaux : le niveau de la session d'interaction, le niveau des tâches et le niveau des actions. Dans les travaux futurs, cette granularité permettra au robot de savoir précisément à quel niveau il doit agir lorsqu'une faible QoI est évaluée. En effet, par exemple, une tâche peut être de mauvaise qualité alors que la session peut encore être considérée comme se déroulant bien.

JAHRVIS a été intégré dans une architecture robotique cognitive et déployé efficacement pour réaliser plusieurs tâches de collaboration et de service dans un environnement réel, comme une tâche de guidage dans un centre commercial et une tâche à deux agents inspirée de la psychologie. Ces tâches ont démontré les capacités du robot en matière de prise de vue, de planification, de représentation des connaissances avec la théorie de l'esprit, de manipulation et de communication.

Mots clés : Interaction Homme-Robot, Action Jointe, Prise de décision, Qualité d'Interaction

Acknowledgments

A faire en dernier :-)

Contents

Acronyms

AEM Action Execution Manager. 69, 71, 73, 90, 99, 105, 106, 107, 109, 112, 113

BDI Belief-Desire-Intention. 5

CM Communication Manager. 69, 71, 73, 103, 104, 107, 108, 109, 110, 112, 113

HAR Human Actions Recognition. 68, 69, 71, 73, 84, 85, 87, 90, 91, 103, 107, 184

HATP Hierarchical Agent-based Task Planner. 44, 78, 79, 92, 93, 94, 98

HATP/EHDA Human Aware Task Planner with Emulation of Human Decisions and Actions. 44, 78, 79, 80, 92, 94, 97, 98, 183

HPM Human Plan Manager. 69, 71, 73, 85, 91, 92, 94, 99, 103, 104, 105, 106, 107, 109

HRI Human Robot Interaction. 9, 12, 42, 63, 74, 79, 171, 176

HTN Hierarchical Task Network. 44, 64, 67, 74, 78, 79, 80, 98

ISM Interaction Session Manager. 69, 71, 73, 82, 99, 109

JAHRVIS Joint Action-based Human-aware supeRVISor. iii, 5, 41, 48, 59, 63, 64, 67, 68, 69, 71, 72, 73, 74, 76, 77, 78, 79, 80, 81, 82, 83, 85, 90, 92, 94, 98, 103, 107, 108, 110, 148, 161, 171, 184

KB Knowledge Base. 59, 72, 73, 74, 76, 80

MuMMER MultiModal Mall Entertainment Robot. 129, 130, 131, 161, 171

NPL Natural Language Processing. 112

QoI Quality of Interaction. 67, 68, 69, 115, 116, 118, 119, 120, 121, 125

REG Referring Expression Generator. 98, 108, 112, 182, 183, 185, 188

RJA ROS-Jason Agent. 58, 59, 68, 69, 72, 73, 77, 82, 84, 85, 92, 98, 103, 107, 108, 110

RPM Robot Plan Manager. 69, 71, 73, 92, 94, 98, 99, 101, 103, 107

SSR Semantic Spatial Representation. 142, 144, 154, 156

ToM Theory of Mind. 8, 9, 10, 34, 92, 103, 174, 175, 180

Introduction

Contributions of the Thesis

List of Publications

Published

- Sarthou, G., Mayima, A., Buisan, G., Belhassein, K., & Clodic, A. (2021, August). The Director Task: a Psychology-Inspired Task to Assess Cognitive and Interactive Robot Architectures. In *2021 30th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN).*

- Mayima, A., Clodic, A., & Alami, R. (2020, August). Toward a Robot Computing an Online Estimation of the Quality of its Interaction with its Human Partner. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* (pp. 291-298).

- Singamaneni, P-T., Mayima, A., Sarthou, G., Sallami, Y., Buisan, G., Y., Belhassein, K., Waldhart, J., & Clodic, A. (2020, March). Guiding Task through Route Description in the MuMMER Project. [Video]. In *HRI '20: ACM/IEEE International Conference on Human-Robot Interaction.* (pp.643-643).

- Belhassein, K., Fernández Castro, V., & Mayima, A. (2020). A Horizontal Approach to Communication for Human-Robot Joint Action: Towards Situated and Sustainable Robotics. In *Culturally Sustainable Social Robotics.* (pp.204-214).

- Foster, M.E., Craenen, B., [...], Mayima, A., [...], Lammi, H., & Tammela, A. (2019, November). MuMMER: Socially Intelligent Human-Robot Interaction in Public Spaces. Artificial Intelligence for Human-Robot Interaction. In *Symposium (AI-HRI), AAAI Fall Symposium Series 2019.*

- Mayima, A., Clodic, A., & Alami, R. (2019, November). Evaluation of the Quality of Interaction from the robot point of view in Human-Robot Interactions. In *The 11th International Conference on Social Robotics (ICSR) (1st Edition of Quality of Interaction in Socially Assistive Robots (QISAR) Workshop).*

Accepted

- Mayima, A., Clodic, A., & Alami, R. Towards robots able to measure in real-time the Quality of Interaction. To be published in *International Journal of Social Robotics.*

Submitted

- Fernández Castro, V., Mayima, A., Belhassein, K., Clodic, A., The Role of Commitments in Socially Appropriate Robotics. Submitted in a volume of the Book *Serie Techno:Phil*.

- Mayima, A., Sarthou, G., Buisan, G., Singamaneni, P-T., Sallami, Y., Belhassein, K., Waldhart, J., Clodic, A., & Alami, R. Direction-giving considered as a Human-Robot Joint Action. Submitted to *User Modeling and User-Adapted Interaction (UMUAI)*.

- Belhassein, K., Fernández Castro, V., Mayima, A., Clodic, A., Pacherie, P., Guidetti, M., Alami, R, & Cochet, H. Addressing joint action challenges in HRI: Insights from psychology and philosophy. Submitted to *Acta Psychologica*.

Part I

Human, Robot and Interaction Models: the Funding Principles of a Decision-Making System for Human-Robot Collaboration

Chapter 1

Lessons from Human-Human models

Contents

This first chapter aims at setting the context for this thesis. First, we present some related works on human-human and human-robot social interactions. These works nourished the thoughts which led to this work. Then, we develop key elements for collaboration such as joint action, commitment and shared plans. Finally, we explore Belief-Desire-Intention (BDI) and cognitive robotic architectures which have inspired us to design our own architecture in which, JAHRVIS — the main contribution of this thesis — endows a robot with the abilities not only to control, but also to evaluate its joint action with a human.

1.1 What is a social interaction?

1.1.1 How to define a social interaction?

First, let's take a look at the dictionary and see how the word *interaction* is defined. According to the Oxford dictionary, an interaction is a "reciprocal action or influence" and more precisely a "communication or direct involvement with someone or something". As for the Cambridge dictionary, it defines it as an occasion when two or more people or things communicate with or react to each other". Those

definitions can give an hint about what it an interaction between humans but they are not specific enough. Now, going through social psychology literature, one of the first attempt to define *social interaction* was by Goffman [111]. He distinguished three basic interaction units: the social occasion, the gathering and the social situation. The social occasion is an event that is temporally and spatially situated in such a way that it forms a unit that can be looked forward and back upon, by participants that are informed by the event (dinner, meeting, sport game...). The gathering refers to any set of two or more individuals who are at the moment in one another's immediate presence. It can be noted that a social occasion may include several gatherings but that gathering do not need social occasions to occur (they can happen in office spaces, street corners, restaurants...). The social situation refers to the full spatial environment that embraces interacting people. It is created as soon as people engage in interaction, when mutual monitoring occurs and ends when the next to the last person leaves. Furthermore, Goffman distinguished between focused and unfocused interaction (gathering). A focused gathering has its members that can come together to sustain a joint focus of visual and cognitive attention and are open to each other for talk. He calls it encounters or engagements. On the other hand, an unfocused gathering has its members present to one another but not engaged together (*e.g.*, persons waiting for a bus). In this same book, Goffman proposed a definition of social encounter: "an occasion of face-to-face interaction, beginning when individuals recognize that they have moved into one another's immediate presence and ending by an appreciated withdrawal from mutual participation".

A couple of years later, Argyle wrote a book entitled Social Interaction [9], where he laid the foundations to understand social interactions. He came to the view that social interaction could be interpreted as a set of social skills, and that it may therefore be possible to train these skills the same way as manual skills are trained. For example, during an encounter between two persons, each must be able to perceive the social cues (verbal or non-verbal signals) of the other which are then filtered through the perspective each has acquired through socialization and experience. The interpretation of context and social cues is then applied to come to a definition of the situation, which in turn guides both behavior and action.

Then, Rummel proposed a definition of a few words: "Social interactions are the acts, actions, or practices of two or more people mutually oriented towards each other's selves, that is, any behavior that tries to affect or take account of each other's subjective experiences or intentions." [238].

The elements brought here, trying to define what is an interaction and more precisely a social interaction, are chosen among a large amount of work. It is possible to find different definitions.

1.1.2 Structure of a social interaction

Most of the research about interaction and social interaction belongs to the field of social psychology. As for the structure of a social interaction, it is more from the

field of Conversation Analysis (CA) which mixes sociology, anthropology, linguistics, speech-communication and psychology.

Robinson makes a review of the work that has been done about *overall structural organization* [234]. Most of the time in the literature, overall structural organization is discussed in terms of "the overall structural organization of entire, single occasions of interaction". Then, the *overall structural organization* term is generally used to talk about one particular (albeit large) unit of interaction. However, many different types of interactional units can have an overall structural organization. For example, Schegloff encouraged to recognize "'overall structural organization' not as something for the unit 'a single conversation' (or encounter, or session, etc.) alone, but for units like turns, actions and courses of action (like answering or telling), sequences, and who knows what else as well" [253]. He also mentioned that every unit of organization should probably have a local organization and a global organization. Here, the term *overall structural organization* refers to "the overall structural organization of entire, single occasions of interaction".

Robinson tells us that this concept has received relatively little analytic attention and thus is still not well understood [234]. Indeed, research has been more focused on analyzing the organization of individual sequences of action such as turn-takings or conversation openings. Several terms have been used to talk about a *supra-sequential coherence*: big package, set of pre-organized sequences, (social) activity, project of activity or plan of action. Sacks gave the following definition for the overall structural organization of single occasions of interaction: it "deals, roughly, with beginnings and endings, and how beginnings work to get from beginnings to something else, and how, from something else, endings are gotten to. And also the relationship - if there is one - between beginnings and endings" [239, p. 157]. Robinson summarized research about the subject by saying that single occasions of interaction (in a generic or context-free sense) are normatively organized as: (1) beginning with an opening (2) ending with a closing and (3) having "something" in between opening and closing" which can be referred to as topics [234].

1.1.2.1 Opening

Openings are used to begin an encounter. One of the main reference on the subject is the work of Schegloff [252]. Openings and related issues vary depending on the nature of interactions. For example, opening of a phone call to a family member or a friend will be organized as follow: (1) summons-answer (the one calling talks first) (2) identification/recognition of each other (3) greetings and (4) how-are-you. Whereas, in primary-care medical visits, opening is sequenced as: (1) greeting (2) securing patients' identities (2) retrieving and reviewing patients' records and (4) embodying readiness (sitting down and facing one another). More examples from the literature can be found in (Robinson, 2012).

Another work, by Kendon [153], focuses on the greeting part, but more precisely the greeting behavior with the associated non-verbal cues. The greeting behavior is divided in three main phases: the distance salutation, the approach and the close

salutation. The distance salutation only occurs if the greeters as far enough such as they need to get closer if they wish to continue the interaction. This phase starts after one or both participants sight one another and at least one of them identifies a wish to engage in a greeting. In case one of the participant has not seen the other one, he signals his presence by vocalizing the other one's name or by clearing his throat. Then, they orient their bodies towards each other and exchange glances in a subtle acknowledgement that the greeting is desired by both. During this phase, people can also wave or give a sign with their head (*e.g.*, nod). The approach is divided into two sub-phases: the distant approach and the final approach. During the distant approach, people tend to look away whereas when the final approach starts (the greeters are 3 meters or less from one another), they look back at each other and, they smile. Finally, there is the close salutation, the most normalized phase of the greeting. It happens when people are 1,5 meters or less from each others. Then, they can have a non-contact close salutation during which people exchange verbal greetings, or they can hand-shake or embrace (or do something else according to their culture). The greeting is over.

1.1.2.2 Topics

Episodes of interaction vary a lot in their contextualized nature, which leads to a large variety of topics and sequences of topics. Interactions that happen in ordinary or institutional contexts can be pre-organized around one or more topics. Robinson gave examples such as an emergency call or an expected call back by a friend to discuss an expected single item of business [234].

1.1.2.3 Closing

Schegloff is one of the reference on closing as well [255]. A closing can be divided into two phases: the topic termination and the leave-taking. The topic termination has a pre-closing statement which signals to the partner the wish to close the conversation. Then, the leave-taking follows the pre-closing statement and its response and, includes the goodbye exchange. Finally, the partners break co-presence, *i.e.*, physically walk apart.[1] In the context of a phone call, Clark and French defined this co-presence breaking as the *contact termination* when people hang up [74].

With regards to non-verbal cues, Knapp *et al.* listed and analyzed them [163]. The more frequent are eye contact breaking, head nodding, leaning toward the partner and positioning in the direction of the way of leaving.

1.2 How do we represent the "other"? - Theory of Mind

Theory of Mind (ToM) refers to the ability to represent others' intentions, beliefs, knowledge, goals, *i.e.*, their *unobservable mental states* [224, 223]. Thus, this con-

[1] It is not explicitly mentioned in [255] but they precise in a footnote that it would not make sense if the parties remain in co-presence after having being through the closing sequence.

cept is related to some presented above. Indeed, common knowledge requires ToM as "to know what another knows and to be capable of making the sorts of inferences required for common knowledge, one must have an understanding of others (or an understanding of a particular person) in terms of thoughts and beliefs" [287, p. 82]. And thus, shared intention, as stated by Pacherie, "having a shared intention typically presupposes cognitively and conceptually demanding theory of mind skills" [218, p. 1817]. Moreover, some authors showed that ToM development and functioning relied on joint attention [274, 58]. Finally, it has been shown that ToM improves joint planning and so increases the ability to cooperate in joint activity [10]. We can note that Westby and Robinson explained that recent research about ToM, showed that ToM is not only to understand what others think, know, believe or intend (cognitive ToM) but that another part of ToM involves thinking about and experiencing the emotions of others (affective ToM) [303]. In this manuscript, we will leave aside the latter.

As for common knowledge (see Section 1.3.3.6), there is an infinite number of levels, or orders to ToM. Most often, the focus is on the first and the second orders. Perner and Wimmer defined the "first-order belief attribution" as the estimation of one's beliefs (*e.g.*, I think she thinks that) and the "second-order belief attribution" as the estimation of what one thinks about what another person is thinking or feeling (*e.g.*, I think she thinks I think that) [220]. Based on the same principle, Flavell *et al.* proposed levels of role-taking where the Level 1 is defined as "S thinks (knows, predicts or whatever) that O has such-and-such belief (attitude, feeling, etc.) about something (X), about S himself, about O himself or about some other individual or group (O_1)" (*e.g.*, "I know how you feel (about something or someone)"). The level 2 is defined as "S thinks that O is aware of (unaware of, dislikes, etc.) S's or O_1's thoughts (feelings, perceptions, etc.) regarding X, S, O or O_1" (*e.g.*, "I'm sure you know what I think about Bill") [97, pp. 49–51].

ToM is also closely related to another notion that we have not mentioned yet but that is of interest in HRI: perspective-taking which is mostly studied in psychology, sociology and neurology. These fields study how people understand each others and refer to it in different ways: social perspective-taking or role-taking, perspective-taking or empathy [84]. Sometimes, ToM and perspective-taking are used interchangeably, as by Charlop-Christy and Daneshvar defining perspective-taking as an elementary aspect of ToM [68] or ToM can be a synonym of "cognitive perspective-taking" [16]. While, some authors, like Westby and Robinson, differentiated them and showed that perspective-taking is an element among others of ToM [303]. Actually, the perspective-taking to which they referred is the one called visual perspective-taking which is a type of perceptual perspective-taking. Indeed, perspective-taking, as ToM has several dimensions. Some authors distinguish between *perceptual perspective-taking*, referring to the inference that one makes regarding another person's visual, auditory, or other perceptual experience, and *conceptual perspective-taking*, referring to the inference that one makes regarding another's internal experience such as his thoughts, desires, attitudes, plans [192]. Other distinguish two other dimensions: "*cognitive perspective-taking* may be de-

fined as the ability to infer the thoughts or beliefs of another agent, while *affective perspective-taking* [or emotional [136]] may be defined as the ability to infer the emotions or feelings of another agent" [123]. According to the task, they can be requirements for ToM [136]. Cognitive perspective-taking is central to communication, particularly in the creation and understanding of referring expression (*i.e.*, a word or phrase to identify an object) [169].

Another element of ToM that we will discuss here is the ability to attribute false belief to others, *i.e.*, make the distinction between the reality and what one can believe about the world [85], as tasks demonstrating this ability have been extensively used to test theory of mind of individuals, such as the task created by Wimmer and Perner [304] and then extended by Baron-Cohen *et al.* which is the most famous false belief task, the Sally–Anne test [304, 18]. In the experiment, children are presented two characters, Sally (who has a basket) and Anne (who has a box). Then, Sally departs, leaving a object A in her basket. While Sally is away, Anne removes the object and hides it in her box. Children are asked to predict, on Sally's return to the room, where Sally will look for the object. Authors used to claim that children being able to answer to this question had ToM whereas others did not. Nowadays, it has been shown that false belief tasks are not enough to assess ToM [33, 302].

1.3 What is a joint action?

Often, multiple concepts are addressed when referring to collaborative tasks: collaboration, cooperation, coordination, joint action, joint activity, shared/joint attention, shared/joint intention, shared plan, shared/common/joint goal, (joint) commitment, engagement, mental states, theory of mind, mutual knowledge... However, many terms and definitions, whether inside a field[2] or between fields do not reach a consensus. This can be quite confusing, especially for roboticists for which it is initially not the range of expertise. Thus, we will first give an overview of the more characteristic definitions of what is Joint Action. Then, we will present a non-exhaustive set of notions related to Joint Action.

A part of this work on Joint Action, realized in the context of the JointAction4HRI project[3], is the result of the collaborative work with Kathleen Belhassein, a PhD student in psychology, and Víctor Fernández Castro, a post-doctoral researcher in philosophy. It has been the subject of a publication in the book Culturally Sustainable Social Robotics [28], a publication under submission at Acta Psychologica [29] and a publication under submission in a volume of the Book Serie Techno:Phil [94].

[2]Here, philosophy, psychology or robotics
[3]https://jointaction4hri.laas.fr/

1.3.1 How to define Joint Action?

An important number of social interactions and encounters are encompassed by the notion of joint action. Broadly considered, joint action is any form of social interaction whereby two agents or more coordinate their actions in order to pursue a joint goal. However, the notion of joint action has particularly been subject to debate in philosophy and psychology. For instance, according to Sebanz *et al.* [261, p. 70], "joint action can be regarded as any form of social interaction whereby two or more individuals coordinate their actions in space and time to bring about a change in the environment."; while other authors [60, 79, 96, 290, 217] resist the idea that instances of mere coordination – *e.g.*, two partners walking side by side – constitute a joint action, considering that it requires some necessary conditions like sharing goals and intentions.

Moreover, while the notion of joint action is used interchangeably with the notion of *collaboration* or *cooperation* for some authors such as Becchio *et al.* [23] and Kobayashi *et al.* [166], other authors establish a hierarchy of interactions depending on the processes involved [6, 66]. According to Amici and Bietti, for example, coordination is a fast low-level process of behavioral matching and interactional synchrony which could, but not necessarily, facilitate middle-level processes like cooperation, collaboration or high-level processes like joint action, which requires other resources like turn-taking and alignment of linguistic resources during dialogue. "To date, however, little is known about the exact way in which coordination, collaboration and cooperation are linked to each other" [6, p. vii]. Looking at the APA dictionary, collaboration is "the act or process of two or more people working together to obtain an outcome desired by all, as in collaborative care and collaborative learning" and "cooperation a process whereby two or more individuals work together toward the attainment of a mutual goal or complementary goals. [...] Often cooperation leads to outcomes [...] but the benefit to each individual is not always obvious" [265]. Thus, here the nuance is in the process benefit and not in the temporal level.

If we look at Sebanz and colleagues definition of joint action, it could be considered as a kind of activity (based on the usual sense of the term activity). Thus, some authors use the concept *joint activity* interchangeably with *joint action* [287, 114] while others see the joint activity composed of joint actions [72, 162]. Clark says that "joint activities advance mostly through joint actions" [72, p .59]. He defines the properties of a joint activity among which there are: it is carried out by 2 or more participants, each participant has a public role or they try to establish and achieve joint goals, and they may have private goals. He also highlights the need for coordination: "What makes an action a joint one, ultimately, is the coordination of individual actions by two or more people. There is coordination of both *content*, what the participants intend to do, and *processes*, the physical and mental systems they recruit in carrying out those intentions" [72, p .59].

Sometimes, it is also possible to come across *collaborative activity* [290], *collaborative task* [50] or *collaborative joint action* [110] (less frequent). Tomasello *et al.*

claimed that "collaborative activities require both an alignment of self with other in order to form the shared goal, and also a differentiation of self from other in order to understand and coordinate the differing but complementary roles in the joint intention" [290, p. 681]. Carpenter, one of the author of this latter article, uses the terms *collaborative activity* and *joint activity* to refer to the same task but giving a social dimension to collaborative activity, being "an end in itself rather than just a means of getting something done" [60, p. 384]. For Pacherie, joint actions are performed by the partners in order to achieve the joint goal of a collaborative activity or joint activity [218].

As we can see, it is not possible to propose a unified definition of these terms based on the literature. In this manuscript, the word joint action will be used to indifferently refer to an activity/task composed of several (joint) actions, *i.e.*, a high level joint action or as a single action, but in both cases it will imply that it is a "social interaction where two or more individuals coordinate their actions in pursuit of a common goal" [64]. We will also use *joint activity* to refer to high level joint actions. Moreover, HRI often refers to collaborative tasks when the robot and the human perform an activity together, thus, we will also use this term which we consider as involving joint actions.

1.3.2 Two possible segmentations around Joint Action

Before going through the mechanisms involved in joint action, we will briefly present two segmentations of joint action: a temporal segmentation, *i.e.*, the different phases a joint action goes through, and a cognitive model for human agency *i.e.*, the different neurocognitive levels that are involved in joint action. It seemed necessary to present these two segmentations as the processes related to joint action described in Section 1.3.3 are sometimes involved in one phase/level but not in another. However, we will not go through these details as it is not necessarily relevant to the rest of the manuscript.

1.3.2.1 Temporal segmentation of Joint Action

As a joint action is a form of social interaction, it can also be divided in three phases as presented in Section 1.1.2: an initiation, a body and a closing [124]. Each phase has a role. First, the initial phase establishes among other things the joint commitment, *i.e.*, who is to participate, in what roles (these can vary during the interaction), what actions are they performed, and when and where they will be performed [73]. Then, in the body participants coordinate to perform their goal. Finally, "to complete a joint action, participants first need to arrive at the mutual conviction that they are both indeed ready to terminate it" [124], if they achieved their goal for example.

At a lower level of joint action, the level of action and not social interaction, joint action can be seen as a process with two phases: planning and execution. Curioni *et al.* proposed a model, specifying what happens in each phase [82]:

- planning:
    - expectations on partner's intentionality
    - selecting action possibilities
    - establishing commitment
- execution:
    - aligning attention to objects and events
    - maintaining commitment

1.3.2.2 Neurocognitive segmentation of Joint Action

To describe the levels of the neurocognitive mechanisms involved in joint action, we will base ourselves on the conceptual framework of action established by Pacherie [216]. This framework is particularly relevant for the rest of manuscript as it has a lot similarities with the three-layered robotic architecture that will be described in Section 3.3, as noticed in [77]. It is based on a dynamic model of intentions and distinguishes:

- A distal intentions level (D-intentions) in charge of the dynamics of decision making, temporal flexibility and high level rational guidance and monitoring of action;
- A proximal intentions level (P-intentions) that inherits a plan from the previous level and whose role is to anchor this plan in the situation of action, this anchoring has to be performed at two levels: temporal anchoring and situational anchoring;
- A motor intentions level (M-intentions) – which encodes the fine-grained details of the action (corresponding to what neuroscientists call motor representations) – is responsible for the precision and smoothness of action execution, and operates at a finer time scale than either D-intentions or P-intentions

This model of action has then been enriched with the specificities of joint action in [217], integrating at each levels the representations and processes associated to the joint action partner. We will not go through the details of it here but they will be mentioned all along Section 1.3.3.

1.3.3 What is necessary for Joint Action?

Leaving aside the debate on the concept of joint action, we aim to focus on the mechanisms that enable the consecution of joint actions. What we found to be the mechanism on which every author (or almost) agrees on to say that it is required for a joint action is the *coordination*. This mechanism itself is supported by other cognitive and sensorimotor processes. Also, philosophers introduced another concept involved in joint action which is the *shared intention*. In Figure 1.1, we made an attempt to represent, in a non-exhaustive way, a number of these processes and how they are connected to each other.

We will first present the concepts of *shared intention* and *joint commitment* and then define the concept of *coordination* and its associated mechanisms.

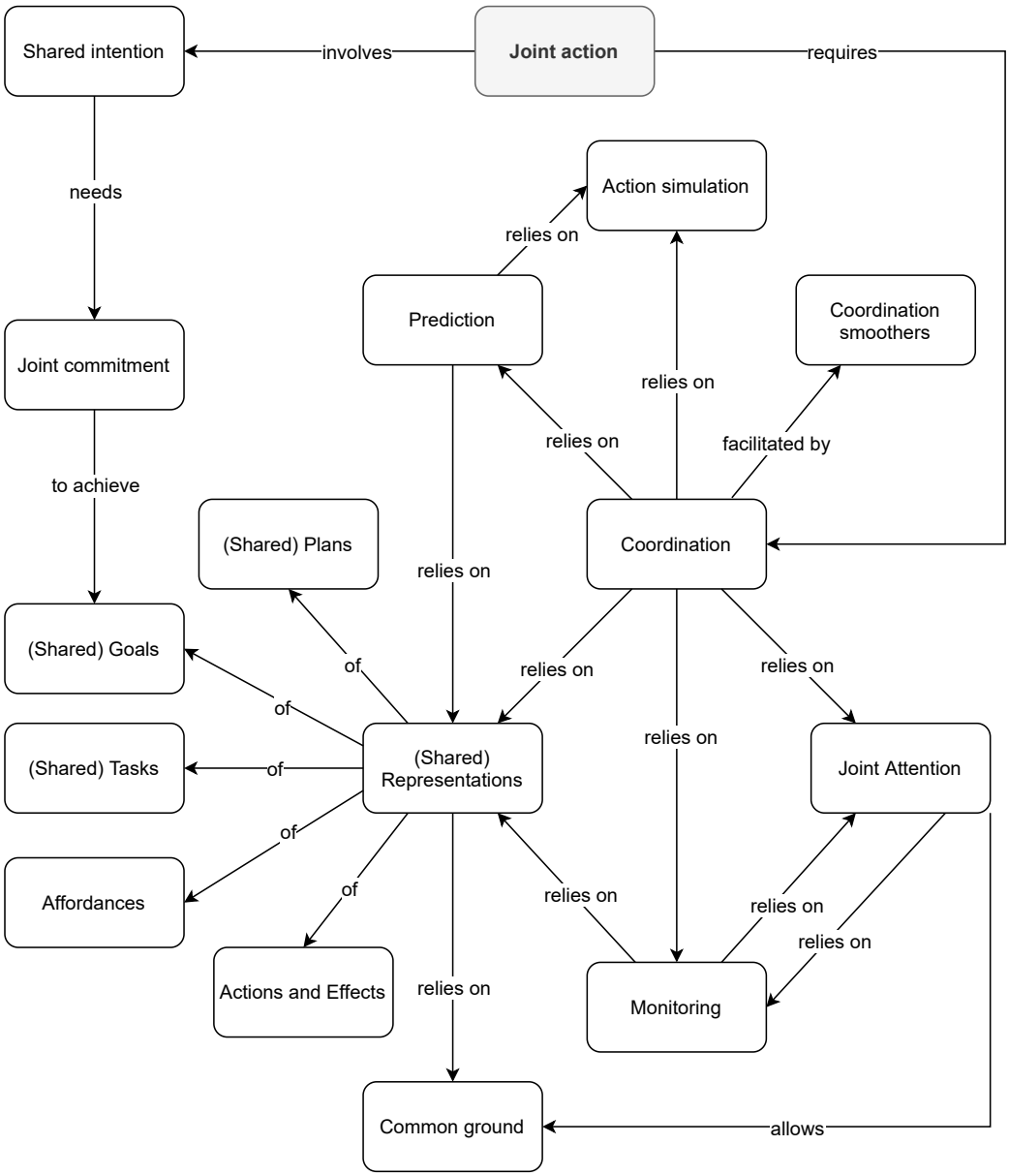


Figure 1.1: Overview of a non-exhaustive set of processes related to joint action

1.3.3.1 Shared Intention

First, before coming to the concept of shared intention, what is an intention? We are going to take a look at three definitions, first from a philosopher (Bratman), then from computer scientists (Cohen and Levesque) and finally from psychologists (Tomasello *et al.*).

Sometimes, we talk about intention to refer to something we do intentionally (action) or to refer to things we intend to do (mental state). Thus, Bratman distinguishes both [38] associating the first possibility to what he calls present-

directed intention (I may intend to start my car now) and the latter to future-directed intention (I may intend to start my car later today). But, "when I am starting my car it may seem natural to say that I no longer intend to start it, I am starting it" [38, p. 379]. He chose to concentrate on future-directed intentions rather than present-directed intentions when referring to intentions.

Cohen and Levesque based their definition of intention on this view of Bratman [78]. They added the notions of commitment and goal: "An intention is defined as a commitment to act in a certain mental state: An agent intends relative to some conditions to do an action just in case she has a persistent goal (relative to that condition) of having done the action, and, moreover, having done it, believing throughout that she is doing it" [79, p. 496].

The definition of Tomasello *et al.* includes the notion of plan since they defined an intention as "a plan of action the organism chooses and commits itself to in pursuit of a *goal*. An intention thus includes both a means (action *plan*) as well as a goal" [290, p. 676].

Now that we have a clearer idea of an intention, we can focus on shared intention. We will start again with Bratman [41]. He considers that two agents have the shared intention to J if and only if:

1. (a) [agent X] intend that [they] J and (b) [agent Y] intend that [they] J.

2. [agent X] intend that [they] J in accordance with and because of 1a, 1b, and meshing subplans of 1a and 1b; [agent Y] intend that [they] J in accordance with and because of 1a, 1b, and meshing subplans of 1a and 1b.

3. 1 and 2 are *common knowledge* between [them].

Tomasello *et al.* refer to shared intentionality ("we" intentionality) and joint intention [290]. For them, it refers to collaborative interactions in which participants have a shared goal (*shared commitment*) and coordinated action roles for pursuing that shared goal. This definition is based on the works of Gilbert [106], Searle [260] and Tuomela [295]. Bratman refered to Searle and Tuomela about their definition of "we-intention", highlighting the difference with shared intention. "we-intention" or "collective intention" are intention of an invidual concerning a group's or collective's activity, and there can be such intention even though there is only one individual (falsely believing others are involved) [41]. Whereas a shared intention necessarily involved at least two persons. Thus, for Tomasello *et al.*, what Bratman calls shared intention, is actually closer to what they call joint intention and not shared intentionality. Indeed, in their view of joint intention, each partner's "representation of the intention [...] contains both self and other", as we can see in their illustration, reproduced in Figure 1.2. For them, joint intention involves *shared goals* with and coordinated action *plans*. Some authors such as Tollefsen use joint intention and shared intention interchangeably [287].

Cohen and Levesque took their inspiration from Bratman's definition of shared intention in their view of joint intention for artificial agents, considering joint in-

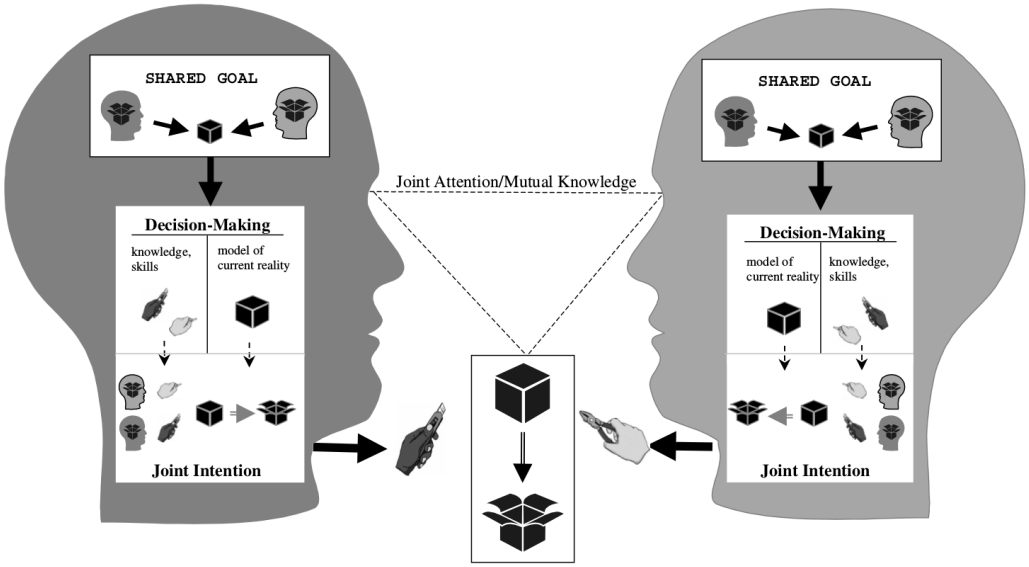


Figure 1.2: Illustrative example of a collaborative activity by Tomasello *et al.* [290]. Here the humans have for shared goal to open the box together. They choose a means to perform it which takes into account the other's capabilities and so form a joint intention.

tention as a future-directed *joint commitment* to perform a collective action while in a joint (or shared) mental state.

1.3.3.2 Joint Commitment

Commitments can be understood as "a triadic relation among two agents and an action, where one of the agents is obligated to perform the action as a result of having given an assurance to the other agent that she would do so, and of the other agent's having acknowledged that assurance under conditions of common knowledge" [198, p. 756]. Commitments are not necessarily established through promises or even explicit verbal communication [179, 249, 272], however, this basic definition allows us to see the fundamental component of a commitment. But why are commitments important for joint action?

In philosophy and psychology, many authors have emphasized the importance of commitments for joint action such as Cohen and Levesque [79], Clark [73], Gilbert [107], Bratman [42], Michael and Pacherie [197], Roth [236] or Siposova *et al.* [272].

For instance, in philosophy, Gilbert [107] and Bratman [42] have largely argued about the requirements for people to establish shared intentions and their role in explaining social coordination. While Bratman has argued that shared intentions can be understood as an aggregation of individual intentions which only requires individual commitments with general standards of rationality, Gilbert has argued that shared intentions are essentially tied to joint commitments. According to her,

two or more persons share an intention to do something if and only if they are jointly committed to intend as a body to do it [107]. In other words, joint actions require the people involved to impose obligations to each other. Further, Roth has argued that joint action requires the participants to be committed to the activity in the Gilbert's sense, which also implies contralateral commitments that hold across the other participants in the shared activity [236]. For instance, if Sue and Jack agree on going for a walk together, they share a commitment to carry out the shared action but also, they assume an individual contralateral commitment to keep pace with each other. In brief, commitments are essential for the establishment of joint and individual intentions during shared activities.

In psychology, several authors, such as Clark [73], Michael *et al.* [199] or Siposova *et al.* [272], have studied how implicit and explicit communication are used to establish commitments and their importance for coordinated actions. For example, Clark has emphasized how partners use communicative exchanges like projective pairs, where one of the participants proposes a particular goal to another (Let's do G! Should we do that?), who then accepts or rejects the proposal [73]. Those exchanges are pervasive in human-human coordinated actions and they serve to negotiate goals, plans and social roles which are translated into an amalgamate of different types of commitments that are necessary for the execution of the general *joint goal*. Michael *et al.* suggest that people often use investment of effort in a task as an implicit cue for making the perceiver aware that we expect him to behave collaboratively which often triggers a sense of commitment that motivates actions [199]. Furthermore, Siposova *et al.* have found that humans use implicit cues like gaze signals to communicate an agreement or commitment to carry out a task their partner intends to perform [272].

To summarize about joint commitment, we retain the words of Gilbert [108, p. 7]: "a joint commitment is a commitment of the two or more people involved. It is, more fully, a commitment by two or more people of the same two or more people", keeping in mind that it "is not a concatenation of personal commitments".

1.3.3.3 Coordination

Coordination is a central mechanism to distinguish individual actions from joint actions. There has been an important deal of conceptual and empirical work investigating this process, such as the one of Knoblich *et al.* [165] and the one of Pacherie [217]. Coordination relies on several mechanisms. They can be non-intentional – sometimes called emergent coordination [165] – such as perception-action matching [37], perception of joint affordances [226] or action simulation [262]. As for intentional coordination – sometimes referred to as planned coordination [165] –, it requires the partners: (i) to represent their own and others' actions, as well as the consequences of these actions, (ii) to represent the hierarchy of sub-goals and sub-tasks of the plan, (iii) to generate predictions of their joint actions, and (iv) to monitor the progress toward the joint goal in order to possibly compensate or help others to achieve their contributions [217]. From Section 1.3.3.4 to Section 1.3.3.10,

we present a non-exhaustive set of mechanisms on which relies coordination. We chose the ones that seemed: to be the most mentioned in the philosophy and psychology literatures, to obtain consensus about their involvement in coordination and to be relevant to Human-Robot Interaction.

1.3.3.4 (Shared) Representations

As stated by Sebanz *et al.*, joint action depends on the ability to share representations [261]. Representation sharing is present at different levels, *i.e.*, agents can share representations of objects, events, actions, goals, plans and tasks [217, 296]. These representations enable, among other things, the prediction (see Section 1.3.3.9) of other's actions.

Representation of (Shared) Tasks A task can be described at multiple grains or levels of abstraction [80], the same action can be described as both 'putting a piece of toast in one's mouth' and 'maintaining an adequate supply of nutrients'. A used definition in psychology is that "a task consists of producing an appropriate action (*e.g.*, conveying to mouth) in response to a stimulus (*e.g.*, toast in a particular context)" [202, p. 1]. Sebanz *et al.* extended this definition with the possibility to execute more than one action when responding to a stimulus [263].

How to share a task? Sebanz *et al.* proposed that "sharing a task representation or corepresenting a task then means that an individual represents at least one rule that states the stimulus conditions under which a coactor should perform a certain action" [263, p. 1235]. In another paper, in the context of joint action, Sebanz *et al.* evoked studies showing the formation of shared representations during collaborative tasks, *i.e.*, an agent knows what the other should do and represents it in a functionally equivalent way to their's own. They concluded that it allowed individuals "to extend the temporal horizon of their action planning, acting in anticipation of others' actions rather than simply responding" [261, p. 73].

Representation of (Shared) Goals Goal has two meanings leading sometimes to ambiguities: the state-to-reach of the environment (external goal) and the mental representation of a desired state (internal goal) [290]. It is interesting to be aware of this double meaning.

Several authors highlighted the need of the representation of a shared goal for joint action such as Pacherie [217], Tomasello *et al.* [290] or Cohen and Levesque [79]. Sometimes they use the terms common goal [259], joint goal or joint persistent goal. Based on Braman [40], Tomasello *et al.* affirmed that "there is a shared goal in the sense that each participant has the goal that we (in mutual knowledge) do X together" and that "each interactant has goals with respect to the other's goals" [290]. Pacherie listed as a condition for joint action that each agent has to represent their goals and their coagents goals [217].

Representation of (Shared) Plans A intention is sometimes defined as a plan of actions that an agent chooses to achieve a given goal [290, 150]. As for shared goals, Tomasello *et al.* and Pacherie highlighted the need for an agent to represent "their own subplans and the meshing parts of the subplans of others, and some of what they represent is to be performed by others" [217, p. 353].

Representations of plans and shared plans are more studied by computer scientists than by philosophers and psychologists, proposing computational models that we will not present here. Grosz and Kraus demonstrated that shared (collaborative) plans should not be treated as the sum of individual plans but as plans necessitating from the agents joint intentions, a mutual belief of how to perform the task and eventually individual or shared plans to perform the task's actions [118].

Representation of Actions and Effects Studies showed that when an agent observes an action, a corresponding representation in their action system is activated [233]. Sebanz *et al.* affirmed that representation sharing is essential to joint action, specially action representations, as "individuals could be 'on the same page' action-wise by sharing representations of actions and their underlying goals" [261, p. 71]. Pacherie evokes the need of agents to have the ability to have not only a representation of actions to be performed, self's and other's, but also their consequences [217].

Affordances The concept of affordances has been introduced in 1966 [4] by Gibson, a psychologist, who presented his theory in [105]. He coined the term to refer to what the environment has to offer to the animal (individual), "what it provides or furnishes, either for good or ill". Then, the term and concept became popular and have been used in other fields than the original one (ecological psychology) such as cognitive psychology, human-computer interaction or design. Osborne discussed in his thesis [215], among other things, the history of the word and the different uses and meaning there are nowadays (see also two reviews on affordances [143, 12]). The definition that is commonly used in HCI/HRI has been introduced by Norman, a design researcher, in 1988 which claimed that "affordance refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used" [1, p. 9]. Thus, it became associated to the term *action possibilities*.

What about affordances when people are in a joint action? Richardson *et al.* showed that when acting together, people take into account not only their motor affordances but also the ones of their partners, helping to decide whether to perform a joint action or an individual action with an object [232]. Thus, Knoblich *et al.* called *common affordance* "when two agents have similar action repertoires and perceive the same object [will likely] engage in similar actions because the object affords the same action for both of them" [165, p. 63], enabling coordination as agents perceive the same objects at the same time. They called *joint affordance*

[4] `https://www.merriam-webster.com/dictionary/affordance`

when "objects have an affordance for two or more people collectively [(a two-handled saw)] which is not necessarily an affordance for any of them individually".

1.3.3.5 Joint Attention

We will start with two complementary definitions of *attention*. The first one is from Tomasello *et al.* who say "attention may thus be thought of as intentional perception (selective attention)" [290]. The second one is from Kaplan and Hafner [150] who define attention as "the temporally-extended process whereby an agent concentrates on some features of the environment to the (relative) exclusion of others". They distinguish two situations for which the process can occur: (1) passive attention when a salient event happens and thus automatically triggers the attention of the agent, and (2) active attention when the agent is involved in an intentionally directed process and must actively select particular features of its environment.

What happens to this process in the context of a joint action? We then talk about *joint attention*. To perform a joint action, partners need a common goal. Indeed, joint action requires that individuals plan and perform their actions according to their predictions about the other's actions to reach this goal. Joint attention is a key feature for this purpose, playing a crucial role in "being and acting together" [288], as it allows the partners to establish and share a perceptual common ground, necessary to initiate the joint action but also for individuals already engaged in a joint action to coordinate successfully.

Despite this agreement to affirm that joint attention is important for joint action, Siposova stated that "there is still surprisingly little agreement on exactly what joint attention is and how it is achieved" [271]. While for some authors, two agents orienting their attention towards the same referent is a sufficient criterion to speak about joint attention [56], others like Pacherie precised that "the phenomenon of joint attention involves more than just two people attending to the same object or event". A classic way to define joint attention is the ability to coordinate our attention to the same object of interest (*e.g.*, as shown by Bakeman and Adamson [13]), enabling us to integrate others' attentional focus and therefore to experience the world together as described by Tomasello [288]. "The attentional focus of the two persons must be truly joint in the sense that both participants are *monitoring* the other's attention to the outside entity" [291, p. 106], thus joint attention cannot exist without mutual knowledge (see Section 1.3.3.6).

To complete this view of joint attention, we can mention Carpenter and Liebal that highlighted the need (1) to develop mutual knowledge of this coordinated attention, and (2) to represent the other agent's intentional states [61] or Kaplan and Hafner that make the notion of *goal* appear in their definition, describing joint attention as (1) a coordinated and collaborative coupling between intentional agents where (2) the goal of each agent is to attend to the same aspect of the environment [150].

Kaplan and Hafner noticed in 2006 that current research in HRI about joint attention tended to focus on "surface behaviors", like simultaneous looking or coordi-

nated behaviors [150] and not what Tomasello and Carpenter called socio-cognitive abilities of shared intentionality [289].

Finally, sometimes it is possible to see references to *shared attention* in the literature, which can be confusing for the reader when no precision or definition is given, differentiating it from *joint attention* or not. Some authors use the two words interchangeably, while some consider that there is a difference between both. There are especially two works mentioning this fact and making a distinction, the one of Emery [92] and a bit later the one of Triesch *et al.* [294]. They define joint attention as two people having the same focus of attention while they define shared attention as a more complex form of communication where each agent knows on what the other agent is focused. We can notice that it is quite similar to the definitions of joint attention we gave above.

1.3.3.6 Common Ground

Common ground, or common knowledge or mutual knowledge, or mutual belief, these are the words to refer to the same general idea – used by some authors interchangeably [71, 72] – but sometimes with nuances like for joint attention. Lewis, a philosopher, claims that a proposition P is commonly known among two agents if the proposition is known by the two agents and both agents know that agent A can draw the same conclusions from P that agent B can and vice-versa. [187]. In another famous formulation of philosophers [258] and psychologists [284], common knowledge must be understood as the recursive belief in which S knows P, Y knows P, S knows that Y knows P, Y knows that S knows P, S knows that Y knows that S knows P, and so on. The subject does not necessarily represent the whole line of reasoning beforehand but should be able to infer it. Thus, we can assume that from the individual point of view, common knowledge or common ground is the information that one may reasonably assume that one and her partner know and they can also know or infer that the other knows. For our purpose, such information may include goals and sub-goals, intentions (see [40]), ways to proceed, facts on the environment (see joint attention in Section 1.3.3.5), appropriate scripts and roles, and any other type of information necessary or relevant for the joint action. Cohen and Levesque, computer scientists, consider mutual knowledge about a *joint persistent goal* P, such as "it is true (and mutual knowledge) that until they come to mutually believe that P is true, that P will never be true, or that [the condition] Q is false, they will continue to mutually believe that they each have P as a weak achievement goal relative to Q and with respect to the team" [79, p. 499].

1.3.3.7 Monitoring

An agent can monitor multiple things related to a task: a goal, an action, a task-progress, mistakes, another agent, an object... Vesper *et al.* showed that an agent typically monitors the task-progress in order to determine whether the current state of the joint action and the desired outcome are aligned [297]. Pacherie named the

monitoring of the progress towards the joint goal as a condition for agent to share a proximal intention [217]. Monitoring the task-progress is not enough. An agent also needs to monitor its partner, especially through joint attention that we presented in Section 1.3.3.5, as attention is a monitoring process and joint attention can be seen as the co-actors' ability to monitor each other's gaze and attentional states [92]. Then, it is also closely related to the shared representations we presented in Section 1.3.3.4. Indeed, shared task representations enables the monitoring of the individual actions [165]. Sebanz *et al.* mentioned "action observation" which seems to be an equivalent to the term monitoring[5]. Action observation and thus monitoring is based on action representations and allows to predict (see Section 1.3.3.9) what others are going to do next [261]. Finally, "monitoring is useful to detect mistakes or unexpected outcomes in one's own or one's partner's performance, enabling one to quickly react and adapt accordingly".

1.3.3.8 Action Simulation

Sebanz *et al.* highlighted the need of action simulation for agents to coordinate [262, 165]. Action simulation is the process allowing an agent to predict the timing and outcomes of the given action, by observing the action and applying predictive models of the action in their motor system. Thus, an agent can predict other agents' actions in real time [305].

1.3.3.9 Predictions

For Pacherie, there are three types of predictions: self-predictions, other-predictions, joint predictions. Self-predictions are the predicted consequences of the agent's own actions. Other-predictions are the actions, goals, motor and proximal intentions of their coagent and their consequences. Joint predictions are the agents' prediction of the joint effects of their own and others' actions. These predictions allow agents to "decide on their next moves, including moves that may involve helping others achieve their contributions to the joint goal (triadic adjustment)" [217, pp. 354-355]. Sebanz *et al.* support the same idea, claiming that predictions, based either on action observation or on shared representations, "allow one to prepare actions in responses to events" [261, p. 73].

1.3.3.10 Coordination Smoothers

Coordination smoothers, as their name implies, are one way to facilitate coordination. They are defined as the changes in an agent own behavior to ease the interaction with another one [297]. For example, an agent may exaggerate their movements, making them easier to predict for their partner. The change of behavior may concern not only one's own behavior but also the use of objects according to their affordance. Coordination smoothers can be produced automatically such as a nod or be intentional [197].

[5] https://www.merriam-webster.com/thesaurus/monitoring

1.4 How do we share information? - Communication

An important part of human psychological devices involved in joint action is communicative, serving different purposes – *e.g.*, negotiating, guiding, questioning [11, 71, 275] and leading to mobilize different types of information. This flexibility allows us to provide information about the relevant objects involved in a task, but also about the emotional or cognitive states of the participants.

Sometimes, people are in situations where social norms, conventions, or scripts are available to regulate our social interactions [251, 7, 63]. For instance, as customers, we usually know how to interact with a waiter in a restaurant because the parties involved know some clear rules of etiquette, social norms and knowledge of how to proceed that regulate the interaction to achieve the joint goal of having a meal. However, even when these rules and norms exist, human interactions require signaling and communicating different types of information regarding the initiation, maintenance, or the exit of joint action, the acknowledgment of roles assignation, or specificities regarding preferences, goals, and subs-tasks.

According to Michael and Pacherie, participants can face three sources of uncertainty during joint action, which can overlap and influence each other [197]. First, motivational uncertainty refers to the uncertainty of not knowing whether or not the partner is motivated to engage in the overall joint action, a particular goal, or sub-goal, or her degree of motivation. Second, instrumental uncertainty refers to the state of not knowing the other participant's instrumental beliefs on how to proceed, *i.e.*, which roles to assume or when and where to act. Finally, common ground uncertainty emerges when instrumental beliefs and motivations are not mutually manifested. Thus, even if the participants share a goal or agree on how to proceed, they might not know that this is the case. Any communicative act or strategy is directed to reduce common ground uncertainty, making mutually manifest a piece of information that can involve instrumental or motivational states, aspects of the environment, goals, or other relevant information for the consecution of the joint action. In a minimal sense, then, communicative strategies can be defined as overt stimuli generated to activate, add up or update the common ground and knowledge related to a particular joint action.

The recognition of the other as a potential partner for joint action can be carried out by verbal and/or non-verbal communicative cues, which can be more or less explicit at different stages of the interaction. The inferential processes at play in such context have originally been explored in the frame of pragmatic theories, in particular through the notions of relevance [275] or Grice's maxims of conversation [116].

Interestingly, humans often establish communicative strategies to facilitate information exchange before the joint action itself. The establishment of *mutual recognition* is fundamental for the initiation of the joint action but also strongly influences its deployment. For instance, establishing mutual recognition facilitates the assignment of roles, which also determines the communicative strategies used during the execution of the action.

Verbal One can engage in communication employing so-called *recognitives* or *observatives*, speech acts whose main function is to call another person's attention upon herself, or other aspects of the context in order to make her aware that recognition is in place.

An example of recognitives is *vocatives*, like greetings that are precisely used to call a person upon herself. Vocatives can enable mutual recognition and facilitate role assignment in some contexts (*e.g.*, "Welcome to our restaurant!" in the previous example). Moreover, vocatives are often followed by other speech acts like questions that can help to set the sub-tasks or goals of the joint action (*e.g.*, "What can I do for you today?"). Another example of recognitives is acknowledgments, whose function is to make the other aware that you recognize or take on what they say (*e.g.*, answering "thank you" to the vocative "welcome"). They allow individuals to acknowledge each other's recognition and to ensure the fact that joint action will take place is mutually shared.

The other types of speech act relevant for mutual recognition are observatives, which serve to identify a potential joint goal by directing the other's attention toward a specific object or event in the near environment. For instance, imagine two hunters searching for prey; when one calls the other "Hey, a deer!", they can start coordinating to capture the animal. Such speech acts can facilitate the recognition of the other as a potential partner for the joint action and then trigger the set of expectations and anticipations necessary to coordinate and perform the action.

Non-verbal *Joint attention* is a kind of communication process, as explained in Section 1.3.3.5, it allows the partners to establish and share a perceptual common ground.

We can also find non-verbal modalities of communication analogous to recognitive or observatives. For instance, communication can stem from subtle cues like the mere reaction to the presence of the other with a frown movement or the search for eye contact. As Brinck and Balkenius [52] argue, by making eye contact, one individual is attending to the other attending to the first, which can implicitly be regarded as a joint commitment to interact in most social contexts. Likewise, acts of acknowledgements can be performed non-verbally as well: people often direct each other's attention toward external objects or events through non-verbal reference, whether it involves vocalizations, gestures, and/or gazing [20, 178, 51]. Non-verbal reference includes four essential actions: a *preparatory behavior* that draws the observer's attention to the sender, a *communicative-intent indicating behavior* to signal the sender's attempt to share attention and interact face-to-face with the observer; a *referential behavior*, to orient the other's attention in the direction of the target object or event; and an *essentially intentional behavior* that orients back the attention to oneself to make sure they understand the act [51, p. 122-123].

To illustrate non-verbal communication during the execution of joint action, we can take the example of the a study on the exaggeration of behavior. In Sacheli *et al.* experiments (see also Vesper and Richardson [298]), for instance, two par-

ticipants had to synchronously grasp an object in an imitative vs. complementary way, each by acting as a Leader or a Follower. The results showed that when acting as leaders, participants tend to give information to their partners about the action to be performed by accentuating some kinematic parameters and reducing the variability of movements, then increasing their predictability by the follower.

1.5 What happens when an agent makes a mistake?

Until now, we have seen the elements facilitating or essential to joint action, but what happens when things go wrong? Things might go wrong because of an error, a mistake, a slip...These words may look like synonym but we can actually make a distinction between them. In this section, we will present a classification that has be done. Then, another subject to take interest to is: there's been something wrong but how to repair now?

1.5.1 Error classification

Reason published a book entitled Human error [230], basing his work, among other things, on Norman [211] and Rasmussen [229]. He classified errors into three categories: *slips*, *lapses* and *mistakes*. All of them are considered as failures. Additionally to these three, he established another kind of failures: the *violations*, which are not errors. He defined slips as attentional failures, *i.e.*, it can be because the agent has been inattentive to the action, not doing the right attentional monitoring (*e.g.*, to take the wrong object and not the one they intended to take). It generally happens with frequently performed actions. Lapses are memory failures, *i.e.*, an agent forget to perform their action (*e.g.*, to go get an object in a shelve and not go back with it because something felt and disturbed the agent). The last type of errors is mistakes, being intentional failures. They happen when an agent choose an action to perform but it is not the appropriate one to reach their goal. Finally, violations are considered as failures but not as errors, being intentional transgression of a rule or a procedure. This classification is illustrated with Figure 1.3.

1.5.2 Repair strategies

In psychology or philosophy, there are not a lot of works on what happens once an error has been made during a joint action. Conversation Analysis investigated repair, which is a way to correct a misunderstanding or an error during an interaction or an action. The ability to engage in repair is essential in interactions. Indeed, errors and misunderstandings are likely to arise and people should find ways to correct them. Generally, they are classified in four categories in CA [254, 306]:

- Self-initiated self-repair: Repair is both initiated and carried out by the responsible of the trouble
- Other-initiated self-repair: The responsible of the trouble takes care of the repair himself but the trouble have been pointed out by the other

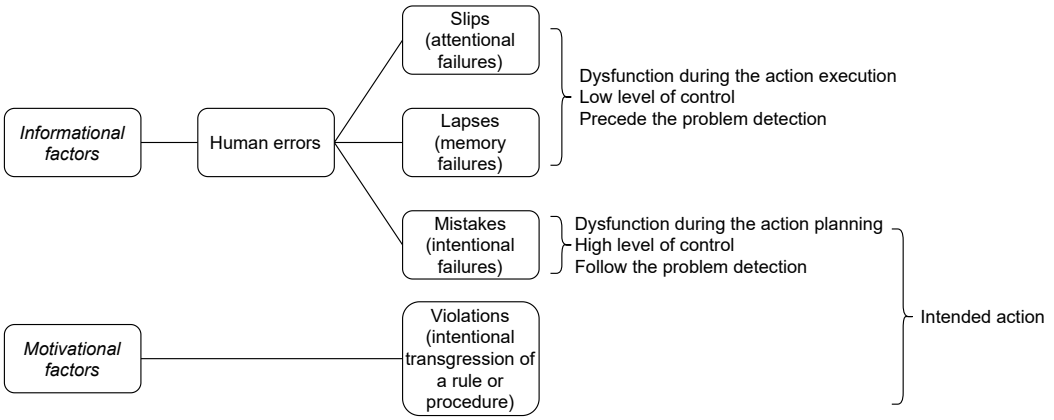


Figure 1.3: Summary and simplification of the failure classification (Generic Error-Modelling System (GEMS) and violations) developed by Reason [230]. (Illustration by Kathleen Belhassein)

- Self-initiated other-repair: The responsible of the trouble signals that a repair is needed and get the other one to repair (*e.g.*, he forgot a name and asks for help to remember)
- Other-initiated other-repair: The one not responsible of the trouble initiates and carries out the repair. This is closest to what is conventionally understood as "correction".

Chapter 2

The "special case" of Human-Robot Interaction

Contents

2.1 Human-Robot Social Interactions

Now that we have seen how social interactions look like when happening between humans, we are going to see the different ways the human-robot interaction field divided and categorized interactions. It is possible to define an interaction according to its length as we will show in Section 2.1.1. Then, inside the interaction, what are the different temporal phases? We will see it in Section 2.1.2. Next, in Section 2.1.3, we will take an interest in a different way to segment interactions: with hierarchical levels. Finally, some authors proposed some interaction patterns, we will present a few in Section 2.1.4.

2.1.1 Interaction lengths

Short-term Interactions Zheng *et al.* defined a *short-term interaction* [308] based on the Unified Theories of Cognition of Newell [209]. A short-term interaction corresponds to the "cognitive band" of cognition, during which they focus on individual utterances and speech acts for interactions that last for tens of seconds. They left aside longer-term interactions that can be in the "rational band" (minutes to hours) or the "social band" (days to months). Gaschler *et al.* deployed a

bartender robot and defined a short-term interaction as being a customer ordering a drink – from the attention request towards the bartender to the closing of interaction by payment and exchange of polite phrases [100]. Iocchi *et al.* used the *short-term* to refer to short interactions and that are focused on only one particular communicative objective, avoiding long and complex interactions [141]. Sanelli *et al.* gave three characteristics to a short-term human-robot interaction: (1) users are not familiar with the robot (2) each interaction happens with a different user (3) interaction is short in time. Then, the robot has not memory of past interactions [241].

Long-term Interactions A survey [182] has been done by Leite *et al.* about long-term human-robot interactions, where long-term means, most of the time, several interactions between the same human and robot. They defined four contexts for which social robots[1] for long-term interaction have been designed: health care and therapy, education, work environment and public spaces, and people's homes.

For example, Kanda *et al.* performed a field trial at an elementary school in Japan for two months [147]. The children were able to interact with the robot for 32 days in total, during 30 minutes after lunch. The robot could switch between one hundred pre-defined behaviors (*e.g.*, hugging, shaking hand or singing) but not all of them were available during the first interactions with a human. Indeed, they had integrated a pseudo-development mechanism, *i.e.*, the more a child interacts with the robot, the more different behaviours the robot displays to that child. Also, the robot confided personal-themed matters to children who have often interacted with it (*e.g.*, "I don't like the cold"). These abilities allowed the robot to maintain the children's interest even after the first week whereas in a first experiment where the robot's behavior was the same all along the two months, most children stopped to interact with the robot from the second week.

In their discussion, Kanda *et al.* raised an interesting question: " How Long Should 'Long-Term' Be?" They found out that some authors consider that two months is a long-term interaction. They also pointed that some Human-Computer Interaction studies on long-term interaction last five weeks. In their survey, Leite *et al.* gave their point of view, which seems well-thought. They argued that it is more important to look at the number of interaction sessions and the length of these sessions (a five minutes-interaction is different from a one hour-interaction). For them, an interaction can be considered as long-term when the user becomes familiarized with the robot to a point that their perception of such robot is not biased by the novelty effect anymore. This definition raises another question: when does user's familiarization with the robot become stable? But we will not discuss it here.

[1] actually, some of the robots featured in the survey are not social robots such as a Roomba or the Personal Exploration Rover (PER)

2.1.2 Interactions divided in phases

Among works on short-term or long-term interactions, some authors divided interactions in phases which have sometimes similarities with the phases of social interactions described in Section 1.1.

Gockley *et al.* divided an interaction in three phases: greeting, core of the interaction and departure [109]. In the greeting phase, Valerie, the robot receptionist, greets people who might be interested in engaging in conversation. Thus, people are classified into "attentional" states:

- present (people a bit far and moving): Valerie doesn't pay attention to them
- attending (people closer): Valerie greets them
- engages (people next to the desk but on the side): Valerie acknowledges their presence but does not expect input from them
- interacting (people in front of the keyboard): Valerie prompts them for input if they are not typing.

In the core of interaction, either Valerie can tell her (fictive) story or chat. Her story is subjective and evolve over time. It is about her social life, her lounge singing career, her therapy business, and her job as a receptionist. Furthermore, Valerie has a chatbot system which is very simple. Finally, inputs from visitors are from a keyboard, for easier control and reliability. Finally, at departure, when a person leaves the "interacting" region, Valerie signals the end of the interaction by saying "goodbye".

Kidd and Breazeal presented robot which is a weigh loss coach [160]. They introduced here the notion of states of relationship. They are three: initial (for the first few days of interaction), normal, repair. According to the state of relationship, the robot answers/questions/speech will not be the same. Kasap and Magnenat-Thalmann designed their system so, to each user, corresponds an interaction session [152]. Each session is composed of four dialogue phases: welcome, warm up, teach and farewell. The system has a memory of users and past interactions. In the memory, is recorded the context (initial state and goal), contents (events) and the outcome (goal succeeded or not). A bit similar to the relationship state defined by Kidd and Breazeal [160], they defined a notion that they call relationship level. It is computed based of the emotional interactions from the episodic memory associated to a user. It influences the mood level of the robot and then the facial expression and the speech.

In the work around the bartender robot [100], they divided the interaction in three phases (or states) but from two different viewpoints, the of the customer and the one of the bartender. From the customer viewpoints, the phases are: (1) attention request towards bartender (2) ordering of one or more beverages, and (3) closing of interaction by payment and exchange of polite phrases. Then, in reaction of each phases, there are the ones from the bartender viewpoint: (1) acknowledging the attention request, (2) serving the ordered drink, and (3) asking for payment. They left open the possibility to have sub-phases inside phases.

We can also find, in Lee *et al.*' work [180], the notion of structure of interaction: interactions start with the vendor identifying the customer, greeting and engaging in small talk with the customer, engaging in the snack transaction, and then enacting social leave-taking.

2.1.3 Hierarchical interactions

Not only, interactions can be divided in phases but also in levels. For example, Dautenhahn and colleagues [83, 212] defined two levels of approach for interactions, a global one and a local one. The *global level approach* defines a unit of interaction as being relatively large (long sequences of interaction or large units of interaction), such as the script for a greeting as described by Kendon in [153]. At this level, an interaction may be seen as a unit similar to a schema or script, in the computer/cognitive science senses of these terms. They named this level of interaction a "Global Interactional Unit (GIU)". Furthermore, a GIU can be divided in phases, each of which has associated behaviors. Behaviors have meaning and their meaning depends on the phase in which they occur, the context (*e.g.*, a 'wave hello' vs. a 'wave goodbye'). Their *local level approach* is a much smaller unit, often as simple as an action and a response to that action. They claimed that this view of interaction has the advantage of greater flexibility and robustness compared to the globally structured view. Flexibility is a result of the possibility of specifying acts that may occur in many global interactional structures. But, as contextual details are ignored, the ability to assign a specific meaning to an action is lost.

In his thesis, Kuo insists about this flexibility and the re-usability [174]. A lower level of design is more appropriate for reuse. For him, a unit of interaction corresponds to an "interaction cue" (or social cue) that a robot can perceive and act upon or express in an interaction. These cues can be verbal, non-verbal, or a combination of both (multi-modal interaction). A complete episode of interaction should be constructed through composition of interaction cues with some common patterns repeated over the course of the interaction (*e.g.*, awareness of human presence).

2.1.4 Patterns of Interaction

Before talking about design patterns or interaction patterns, Goffman argued that human interactions follow a specific "order" and characterized a number of patterns in which people interact, such as how greetings unfold and how people leave an interaction [112].

Kahn *et al.* introduced design patterns [145], that they will later called interaction patterns in [146], inspired from computer science. They proposed rules to follow using them and eight patterns. The two main ideas to retain is that a sequence of patterns has to be well ordered and that patterns can be hierarchical. The 8 patterns:

- The initial introduction: largely scripted, conventionally-established verbal and behavioral repertoire to recognize the other, inquire politely about the

other, engage in some physical acknowledgment (*e.g.*, handshake)
- Didactic communication: one-way communication of information
- In motion together: walk together
- Personal interests and history: sharing of personal interests and history with others
- Recovering from mistakes: creates the potential for both parties to maintain a social affiliation following the mistake
- Reciprocal turn-taking in game contextual: taking turns with one another when playing games
- Physical intimacy: to engage in holding or touching or embracing
- Claiming unfair treatment or wrongful harms: allows to make claim to its moral standing

Following the same idea and going further, Sauppé and Mutlu introduced the interaction blocks [248]. Compared to Kahn's work, they created a pattern language and a tool/environment to design human-robot interaction. To conceive their patterns, they collected and analyzed data from 5 kinds of interaction scenarios: Conversation, Collaboration, Instruction, Interview and Storytelling. Then, they identified common interaction structures, which served as "design interaction patterns":

- Introductory monologue: A short introduction can be used to introduce other participants to a scenario by giving an overview of the remainder of the interaction or it can be a greeting for example.
- Question and Answer: A question is a sentence meant to elicit information from other participants. An answer is the response to a question that aims to satisfy the questioning participant's curiosity.
- Generic Comment and Personal Comment: A comment is a short statement offering the speaker's opinion. Comments are either generic (*e.g.*, , "Wow") or personal (*e.g.*, , "I tried that and didn't like it").
- Monologue and Generic Comment: A monologue is a longer form of speech during which no response is expected.(*e.g.*, telling of a story). Although monologues expect no response, listeners may occasionally offer unsolicited commentary.
- Instruction and Action: An instruction is a command offered by one participant to direct the actions of another participant. The proper response to this instruction is often an action, although the action might follow the instruction with a delay depending on whether it is an appropriate time to perform that action
- Finished Comment: Upon the completion of the goals of the scenario, one or more of the participants will note that the scenario is completed by offering a finished comment.
- Wait: One pattern implicit in all scenarios involving two or more participants is the wait pattern.

Finally, they designed a software to easily implement those patterns in a robot.

In his thesis, Kuo criticizes Kahn's work [174]. He says that these patterns involve sequences of interaction cues and should be decomposed to a lower level for detailed design and reuse. He proposed his own patterns:

- Human presence detection: detect when there is a person who might be interested in
- Showing interest for interaction: express the robot's awareness of a user's presence around it and its interest and willingness to interact
- User's attention on the robot: Know when a user is paying attention to the robot in an interaction and its information on its screen
- User identification by face: Provides the fundamental block for personal service and social interaction by recognizing the human counterpart in an interaction

He checked the validity of his patterns with the analysis of Problem statement, Context of Use, Interaction Modality, Combination with Other Patterns, Technical Performance and Limitations, User Feedback and User's Perception, Resulting Interactive Behavior.

Finally, Peltason and Wrede also based their work on design patterns from computer science, specifically applied to dialogue here [219]. To name a few of them: Simple action request, Interaction opening, Interaction closing, Clarification. During interaction, the registered patterns are employed in a flexible way by admitting that patterns can be interrupted by other patterns and possibly resumed later which leads to interleaving patterns. By default, simpler patterns are permitted to be nested within temporally extended patterns.

2.2 Human-Robot Interaction and Joint Action

The study of human-human joint action is important to understand how to make robots better companions and partners for humans. However, it does not mean that they should imitate humans, as they are machines, they have their own abilities and have to develop their own strategies [36] (*e.g.*, displaying an arrow on the floor while navigating [65, 81]).

In the context of the JointAction4HRI project, a non-exhaustive review of existing robotic systems integrating but also recognizing in humans joint action mechanisms has been done, focusing on joint attention, communication to facilitate coordination, repairs strategies and commitments.

2.2.1 Joint Attention in HRI

Joint attention is essential to joint action (see Section 1.3.3.5). Some authors showed that a robot initiating (*i.e.*, triggering the attention focus of the partner on the object of interest) [137], responding to (*i.e.*, gaze following of the partner's gaze or

gesture) [307], and ensuring joint attention (*i.e.*, monitoring of the other's attention) [134] improves the task performance and is perceived as more natural. Thus, human pointing gesture recognition as been investigated such as in [210] or eye-gaze signaling(*e.g.*, [277] or see review [2]).

2.2.2 Communication to Facilitate Coordination in HRI

As seen in Section 1.4, communication is important for joint action. It is useful to negotiate, guide, question or realign the beliefs between agents as divergences might occur [79]. Here, we will see how robots can do to communicate and understand communications about: (1) their internal state or the human partner's one, and (2) intentions.

Internal state communication – Expressions It is not always obvious for the human to know what the robot is "thinking", *i.e.*, to know in what state is the robot. The robot also needs to be able to recognize human internal states. Roboticists developed different ways to do so. We found that robot communicating their internal state using lights, dialogue, gestures/moves or facial expressions has been developed. Some of them are also able to analyze human face or voice to detect their emotion or expression. Kim and Kwon designed a robot using all these features to generate expressions according to its knowledge about the task execution state [161]. Expressions are generated based on as set of criteria. For example, when the robot computes that it is in an unexpected state, it generates surprise. Moreover, they endowed the robot the ability to discriminate between the human partner's happiness, sadness and anger. In the same spirit, we can find a robot recognizing and generating expressions through voice, in relation to the task state and goal [256]. Finally, we have to mention the work of Breazeal which investigated a lot display of emotions/expressions in human-robot interaction. We can distinguish two types of communication: (1) a robot, which as a caregiver, has a motivation system in order to regulate the interaction intensity of its caregiver by expressing eight emotions with facial expressions [49, 48], (2) a robot has the ability to recognize four communication intents (approval, attentional bid, prohibition, soothing) and to react to them through speech [46, 47].

Communication of intentions As explained in Section 1.3.3.10, coordination smoothers facilitate the prediction and legibility of a partner's action. Investigation about how the robot could communicate its intention during navigation has been done. Some authors chose to have the robot communicating its intention using lights [279], comparing this method with a communication using the head orientation and finding it better [194]. Others worked on make the robot navigation legible, improving its predictability by the human [90, 3]. A last method to communication navigation intention is by projecting arrows on the floor as well as a map [65, 81]. Not only robot navigation should be legible but also its gestures, when handing over an object to the human [273], or opening a door [281] for example. Finally, a lot

of works can be found on human gestures recognition so the robot could prediction human's action intentions (*e.g.*, [19, 67]).

2.2.3 Theory of Mind in HRI

Theory of Mind (ToM) is related to joint action as shown in Section 1.2. One of the first work to bind robotics and ToM is Scassellati [250]. He proposed a model of a ToM implementation for a robot, inspired by two models from psychology: Leslie's Model of Theory of Mind [186] and Baron-Cohen's Model of Theory of Mind [17]. His model focused on two abilities: to make the distinction between animate and inanimate visual stimuli (following Leslie's perceptual world division into animate and inanimate spheres), and to identify gaze direction (enabling the shared attention mechanism as emphasized by Baron-Cohen).

Over the years, others tried to tackle this issue, especially focusing on perspective-taking abilities. For example, Hiatt and colleagues designed and implemented a model simulating a human with the ability to deal with false beliefs [129]. They demonstrated it with the Sally–Anne test presented in Section 1.2. Milliez *et al.* endowed a robot with the ability to pass the Sally–Anne test, constructing a semantic representation of the world based on its estimation of the human's point of view [201].

Perspective-taking abilities have been used in robotics for several purposes. Berlin *et al.* presented a simulated robot able to take the visual perspective of a human teacher (the virtual camera) and showed how this ability could be used for learning in human-robot interaction [31]. Hiatt *et al.* presented a humanoid robot reasoning on possible beliefs the human partner could have endowing the robot with the ability to deal with uncertainty about the estimation of the human's beliefs [127]. In another work, perspective-taking allows the robot to solve ambiguous references to an object [235]. Others make use of perspective-taking to improve human action recognition [144]. Endowing a robot with a perspective-taking ability can also serve to support the implementation of an autobiographical memory (a meaningful stored knowledge acquired during interactions) [222]. Finally, it can also be a way to help the robot to elaborate plans, adding communication actions to solve divergent beliefs [301], to explain plans to the human with a level of details depending on their knowledge [200] or to manage shared plans execution [87].

2.2.4 Failures in HRI

Honig and Oron-Gilad studied the different types of failures that could happen during a human-robot interaction and proposed a classification of these [132] based on a meticulous review, illustrated with Figure 2.1.

2.2.4.1 Communicating about Failures

When something unexpected happens during the interaction, either because of a human failure, a robot failure or an external event, robots might need to communicate

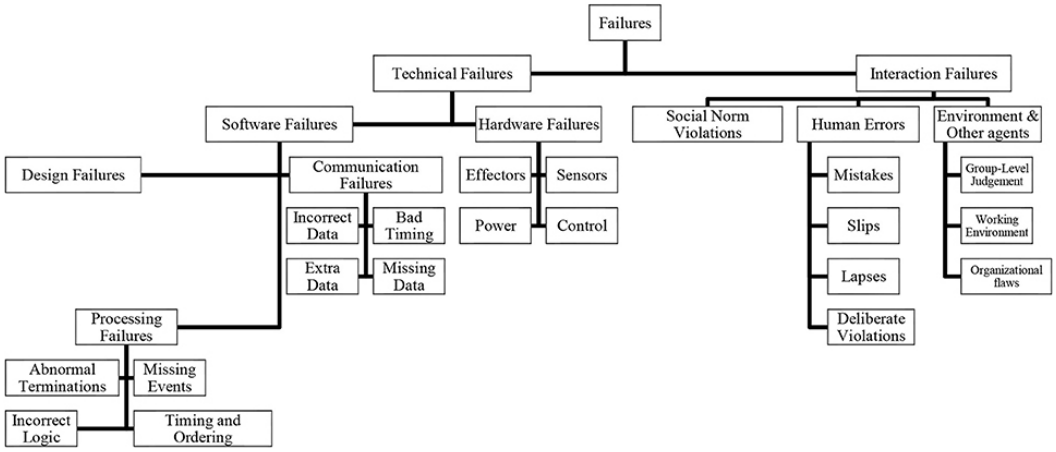


Figure 2.1: Classification of failures in HRI, realized by Honig and Oron-Gilad [132].

about it. We saw three most common ways to communicate about a failure: facial expressions and speech. We can notice a less common way proposed by Kwon *et al.* which developed a robot communicating about its incapability to execute a manipulation tasks thanks to the execution of an "attempt motion", expressing what it cannot do and why to its human partner (*e.g.*, lifting its elbow to communicate that it is trying to lift the cup, but the cup is too heavy for it) [175].

Facial expressions Breazeal and colleagues investigated the expression of the robot's confusion through facial expressions. For example, in [46, 47] humans take into account the robot expressive feedback to assess when the robot "understand" them. If the wrong expression appeared, they often speak in an exaggerated way to correct the "misunderstanding". Hamacher *et al.* developed a robot displaying a facial expression after having dropped an egg when preparing an omelet with a human [120]. Reyes *et al.* proposed a robot displaying a negative facial expression when an error occurs [231]. In work by Silva *et al.*, the robot decision-making and error detection/handling processes are influenced by the perceived human emotions and the robot can display facial expressions when the human persists in a error [269].

Speech Most of the time, when speech is involved, it is not only to communicate about the failure but also to initiate a repair strategy.

2.2.4.2 Robot Repair Strategies

To communicate about a failure is not enough to be able to continue the task or the interaction. A collaborative robot needs repair strategies. Some authors proposed simple strategies while some others developed more complex ones. In [188], when the robot detects a failure in its understanding of the human utterance and gesture, it triggers its repair mechanism that leads the robot to ask questions to the human in order to help it disambiguate the utterance/gesture. In [203], there were two

possibilities when a failure happened: the human had an “assistance opportunity” (*i.e.*, failure case that caused no risk to person or property) before the failure occurrence or not. People were more willing to help the robot in case of failure with an assistance opportunity. Knepper *et al.* proposed a robot, when encountering a failure, able to request help from a human – which might not be aware of the context. After receiving help, it resumes its autonomous task execution [164]. Lee *et al.* compared three recovering strategies after a robot failure (but based on videos and not a real robot) to see which one was preferred by the humans: (1) apology (*i.e.*, robot apologies for service failure), (2) compensation (*i.e.*, robot provides compensation, such as an exchange, a refund, or a discount coupon), (3) option (*i.e.*, robot provides human with alternative actions to achieve their goals) [181]. All strategies had a positive effect (even if not the same). Spexard *et al.* developed three repair strategies according to the type of failure for their robot: (1) when the robot encounters an internal issue, it informs the human about the break-down and asking them a reboot or to contact a technician, (2) it can generate appropriate speech related to error messages from its sensor, *e.g.*, the robot informs the human of the reason why it can not move and asks them for help, and (3) the robot asks for a reset if it thinks that the information it has about a human does not seem to be right [276]. Mutlu *et al.* performed a user study to compare three repair strategies in a task where the robot gives instructions to the human in an assembly task [208]. The strategies are: (1) “no repair” (*i.e.*, the robot detects an error but waits without answering to the human’s questions), (2) “simple repair” (*i.e.*, the robot answers by yes or no to the human yes/no questions, for other questions it repeats the instruction), and (3) “humanlike-repair” (*i.e.*, the robot gives the appropriate information to human, triggered either by a human request, or failure or hesitancy detection). The last strategy was the preferred one by the participants.

Part II

The Challenge of Social Interaction Management

Chapter 3

Architectures for Collaborative Robots, Decision and Execution

Contents

Robots are machine which need to perceive, decide and act. There are multiple ways to endow a robot with such abilities, with different levels of complexity. When a robot has a complex and generic software architecture, based on models which might be inspired from other fields like psychology, philosophy, neurology, it is referred to as cognitive robot or autonomous robots or intelligent robot...We are interested in such architectures but designed to be implemented in collaborative robots. And, we take an interest in a particular function of these architecture: the decision-making, the supervision of the task, of the interaction.

3.1 Existing Architectures for Collaborative Robots

"An integrated cognitive architecture can be defined as a single system that is capable of producing all aspects of behaviour, while remaining constant across various domains and knowledge bases" [70, p. 104]. Kotseruba and Tsotsos reviewed cognitive architectures starting 40 years ago until nowadays. They accounted around three hundred of them and chose to focus their review on 84 [168]. However, the term *cognitive architecture* often refers to an architecture modeling human cognition [133] and what interest us is to endow robots with cognitive and interactive abilities, not always basing ourselves on human cognition.

Some cognitive architectures such as ACT-R has been adapted for human-robot interaction (ACT-R/E) [292]. The architecture aims at simulating how humans think, perceive and act in the world, strongly based on theory of mind. It is interesting but to understand humans is not enough to make the robot a good

collaborators for them, as it lacks abilities concerning the human-aware task and action execution.

A very complete architecture, CRAM, dealing with problems such as manipulation, perception, plans or beliefs management has been developed by Beetz *et al.* [25]. However, this architecture is more designed for a robot acting alone than a robot acting in collaboration with a human.

The work of Scheutz and colleagues is compelling, as they proposed a generic architecture, DIARC, for cognitive robots collaborating with humans [256, 257]. In this context, it handles perception, dialogue and different kind of actions. But, the architecture lacks real modeling and awareness of the human at each level.

Another architecture worth to be mentioned is the DAC-h3 architecture by Moulin-Frier, Fischer *et al.*, inspired from biology [206]. It is designed for a robot maintaining social interactions with humans, able to tell narratives and to acquire knowledge thanks to its interactions with humans. As it is mainly dedicated to knowledge acquisition and expression, it lacks planning and execution abilities.

Finally, there is the architecture developed and implemented by Lemaignan and colleagues for collaborative robots. All deliberative components of the architecture are human-aware [185], *i.e.*, all of components except the sensorimotor layer. This architecture is based on the philosophical BDI model developed by Bratman [39, 43]. It has 3 main concepts defined as the following in computer science:

- *Beliefs*: They are a representation of the agent's knowledge about the world. "[They] can be viewed as the informative component of system state" [228, p. 313]. It is not the word "knowledge" that has been chosen to define this concept because what the agent perceives of the environment is in fact the likely state of the environment. There is no certainty, its sensors are not accurate or could malfunction. This way of distinguish knowledge and beliefs is one that can be found in the literature of distributed computing [177].
- *Desires*[1]: They are a representation of the motivational state of the system. They provide "information about the objectives to be accomplished or, more generally, what priorities or payoffs are associated with the various current objectives" [228].
- *Intentions*: They are a representation of the currently chosen course of action (plan). It is the deliberative component of the system. The selected course(s) of action are determined with a deliberative function, according to the beliefs and desires [228].

[1]In one of the first implementation, PRS, "Goals" notion was used instead of "Desires" [102], then they use it in a interchangeable way in [101] and finally choose "Desires" [228] with the definition given in the AI literature, *e.g.*, desires can be many at any instant and may be mutually incompatible. Therefore, a goal will be a chosen desire [78] and concurrent goals are consistent.

3.2 Lack and needs... TO BE DONE

3.3 The new LAAS Architecture... voir si votre archi a un nom

3.3.1 Specificities...

3.3.2 Overall architecture explanation

mention three layered archi Pacherie ref chap1

In this section, we will shortly present an overview of the robotic architecture that has been developed inside the RIS team of LAAS-CNRS. The purpose of this overview is inform the reader about the inputs available for JAHRVIS. Two instantiations of this architecture for two different tasks will be described in Chapter 8 and Chapter 9. This architecture model, shown in Figure 3.1, has been inspired by the architecture developed by Lemaignan and colleagues [185], mentioned in Section 3.1. All communication between the components goes through ROS [225].

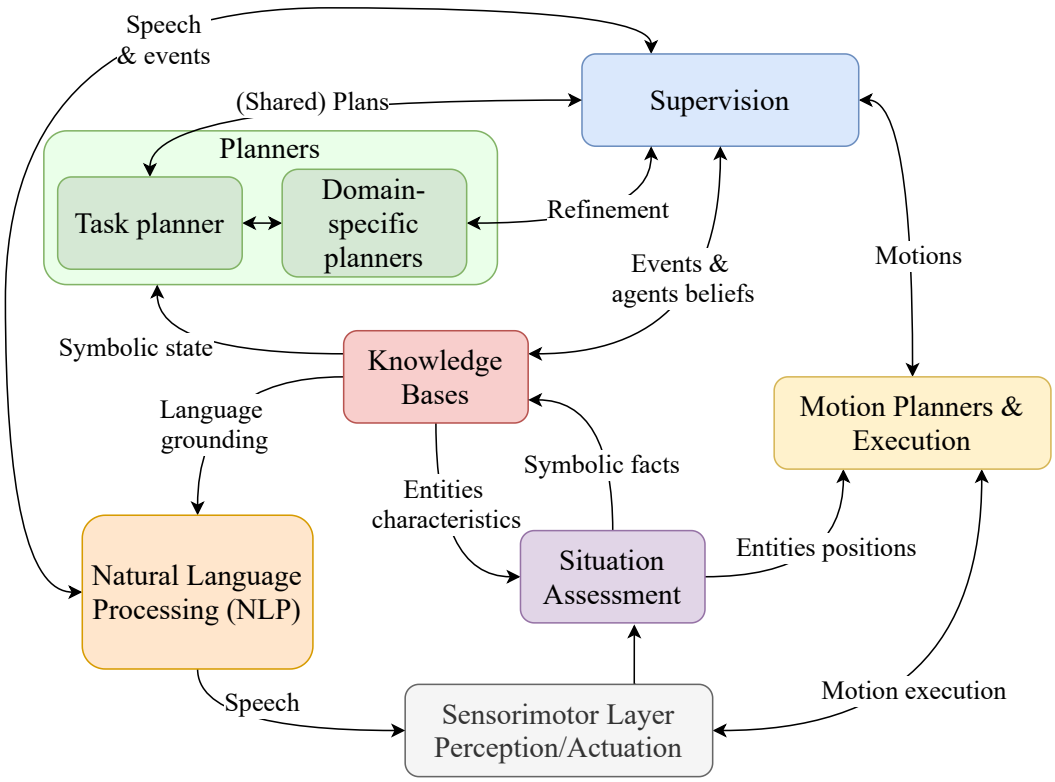


Figure 3.1: Overview of the RIS architecture

3.3.3 Architecture components

3.3.3.1 Situation Assessment

The Situation Assessment has two roles:

1. to gather different perceptual information and build a geometric representation of the world (*i.e.*, elements have associated 3D coordinates), composed of objects and agents; from this world representation, the module runs reasoning processes to interpret it in terms of symbolic statements between the objects themselves and between the involved agents and the objects. Such component can be implemented with frameworks like Toaster [201] or Underworld [184].

2. estimate the human's perspective and build an estimation of their world representation; it is the first step allowing to implement the theory of mind principles (see Section 1.2).

Thus, the Situation Assessment outputs data, which we call *symbolic facts*, such as *isOnTopOf*(cube_1, cube_3) or *isReachableBy*(cube_2, human_0). The first element of this triplet is called the property, the second one is the subject and the last one is the object.

3.3.3.2 Knowledge Bases

Knowledge management is central in a robotic architecture. As this architecture is specific to HRI, the knowledge handling should be as well. Indeed, during an interaction, belief divergences may arise between agents and thus how this should be represented? Each agent, human and robot, should have their own knowledge base. Software developed by Guillaume Sarthou with whom we collaborated closely allow this. They are two of them: Ontologenius[2] [244] for the semantic knowledge and Mementar[3] for the episodic one. They are fully adapted to HRI applications by representing the robot's knowledge and the estimation of the partners' knowledge separately, which refers to the psychological concept of the "self-other distinction" as coined in joint action studies [217].

Semantic knowledge base stores common-sense knowledge based on three concepts, in the form of an ontology[4]: (1) classes representing the possible types of entities known by the agent (*e.g.*, Cube is a class inheriting from the class Pickable), (2) properties which can denote both the attributes of objects (*e.g.*, the color) and the relations between the objects (*e.g.*, which object is on which other one), and (3) object instantiations also called individuals (*e.g.*, cube_3 is an object present in the environment, of the class Cube).

It is in charge of representing the environment elements meaning, the objects' and agents' types (*e.g.*, a cube is an object), their applicable properties (*e.g.*, cube_1

[2] `https://github.com/sarthou/ontologenius`
[3] `https://github.com/sarthou/mementar`
[4] An ontology is a way to represent knowledge

has color blue), the descriptions and parameters of the actions, a part of the language model with verbs or pronouns, and their names in natural language.

Besides, it is also in charge of representing the current symbolic world-state (the computed facts, *e.g.*, *isOnTof*(cube_2, table_1)) and thus the instantiation of the concepts in terms of physical (*e.g.*, this particular block) or abstract (*e.g.*, this particular action instance) entities. Moreover, it reasons on it, making deductions and links between facts, creating new ones (*e.g.*, after receiving *isOnTof*(cube_2, table_1) and it computes *isUnder*(table_1, cube_2)). Finally, it stores knowledge about activities grounded in space and time (*e.g.*, object_1245 has been put on the table_2 by robot (action ID 475)).

To access to the knowledge stored in Ontologenius, the Supervisor can make a request to know if a given fact exists or ask an information about a class, a property or an object instantiation (*e.g.*, the Supervisor can ask the human understandable name of pick_action which is "pick"). Another way is to subscribe to updates (addition or deletion) for given facts (*i.e.*, facts necessary to the task or the Supervisor functioning). It is useful for keeping updates about the environment and avoid to be snowed under too much data. For example, the Supervisor can ask to receive every update (addition or deletion) of any fact belonging to the type *isOnTopOf*(Cube, Table). In this case, it will receive the addition of *isOnTopOf*(cube_2, table_1) but not of *isOnTopOf*(spoon, table_1). It is possible to either specify the class or individual of the subjects/objects that should be concerned by the subscription, or to receive every facts (*e.g.*, it can subscribe to receive additions of the human looking at the robot `[add]human_0|isLookingAt|robot` or to receive all updates about every objects that the human looks at `[?]human_0|isLookingAt|?`). The way the Supervisor chooses which fact it should receive is described in Chapter 6.

Episodic knowledge base is represented as a timeline, keeping track of the symbolic facts computed over time (*e.g.*, action ID 475 started at 3286 seconds and was over at 3290 seconds), either by the Situation Assessment, the Semantic Knowledge Base or the Supervisor.

3.3.3.3 Human-Aware Motion Planners and Execution

The motion planners allow the robot to execute human-aware motion actions. According to the task needs, several planners might be involved for a same task. Indeed, in a task requiring object manipulation, the robot will need a motion planner able to plan for pick, place and drop actions, such as MoveIt[5] or GTP which is human-aware [300], and a home-made software handling the execution these trajectories[6]. Moreover, in collaborative tasks, an agent might be led to hand over an object to their partner, in this case could be used a dedicated planner [190]. Finally, the robot might need to move in the environment, but when moving in a

[5] `https://moveit.ros.org/`

[6] `https://github.com/YannickRiou/pr2_mtc`

environment with humans, it should navigate being aware of them for safety and legibility. Thus, the robotic architecture should integrate co-navigation planner and executor such as HATEB-2 [270].

These planners produce trajectories and moves on request of the Supervisor. During execution, they sends feedbacks about the state execution, in this way the Supervisor can receive data about something going wrong or the estimated remaining time of execution.

3.3.3.4 Human-Aware Task Planner

We can distinguish two situations in which a collaborative robot needs a human-aware task planner: (1) when it performs a task on its own but a human is nearby and so it should consider potential conflicting actions with them, and (2) when it performs a task with a human. In the first situation, it might consider asking the human's help whereas in the second one it needs to plan for both agents' actions.

The human-aware task planners generate symbolic shared plans in which each agent, human and robot, has actions of the task assigned to them, depending on criteria such as programmer-defined costs, minimization of the human's effort, and their preferences and comfort. Such plans are constructed to be as respectful as possible of social constraints which benefits to the also human-aware Supervisor. However, the human is neither an agent that the planner can directly control or an agent that will know the complete plan. Thus, it allows the robot to plan by emulating the human decision, action, and reaction processes.

We worked with two human-aware task planners: Hierarchical Agent-based Task Planner (HATP) [176] and Human Aware Task Planner with Emulation of Human Decisions and Actions (HATP/EHDA) [53].

The Hierarchical Agent-based Task Planner (HATP) proposes a hierarchical approach to multi agents task planning. This Hierarchical Task Network (HTN)-based planner is able to elaborate a multi agents plan based on a single HTN tree. HTN planning aims at decomposing abstract tasks into primitive tasks by choosing from a list of available context-dependent refinements for each abstract task, ensuring that preconditions and effects of refined primitives tasks are satisfied throughout the created plan. This formalism is suitable for human-robot interactions as it allows the robot to communicate about the plan more easily. HATP has been specially designed to integrate a number of features that are meant to promote the synthesis of plans that are acceptable by humans and easily if not trivially understandable by them. It allows to specify the humans and robot capabilities in terms of actions they can execute. Several aspects such as human preferences and comfort, estimation of human effort to achieve a task in a given context and "social rules" are used in a cost-based approach to build "sufficiently good" human-robot shared plans.

The Human Aware Task Planner with Emulation of Human Decisions and Actions (HATP/EHDA) also proposes a hierarchical approach to multi agents task planning. This approach relies on a representation of each agent considered by the planner, with their own beliefs, agenda, stream of execution and action model. The

action models are represented as HTNs which are explored consecutively yet differently if the agent is controllable (robots) or uncontrollable but rational (humans). We highlight two main advantages over HATP. The first one is the computation of conditional plans, allowing to anticipate situations where the human may perform multiple different actions equally making the plan progressing, or may decide to act or wait for the robot to act. Then, the decision of which branch of the plan to follow is postponed at execution time and handled by the Supervisor. Another advantage is to explicitly represent robot to human communication needs for beliefs alignment, goal sharing or action requests. Indeed, HATP generates plans in which it is unknown what needs to be communicated to the human or what can be easily guessed (*i.e.*, which is predictable) by the human. Finally, like in HATP, it is possible to define "social costs functions". By doing so, the planner can penalize non-acceptable sequence of robot actions (*e.g.*, serving a meal just after taking out the trash) or non-satisfactory human required contribution (*e.g.*, or requesting the human to perform small tasks multiple times instead of giving the big picture of the real task to perform).

3.3.3.5 Supervision

The Supervision is the puppet master of the system, embedding the robot high-level decisions, controlling its behavior and trying to react to contingencies, always considering the human it is interacting with. It is not standalone, relying on the components described above to be able to take decisions, be aware of the environment and make the robot moves.

After giving an overview of the components on which the supervision relies, we present the context in which we place ourselves for human-robot collaboration.

Chapter 4

The central and pivotal role of Supervision

Contents

4.1 State of the art

The supervision component is the binder of a robotic architecture. Without it, there is no task, no interaction happening. Indeed, what we define by 'supervision' is the higher level of the architecture, the process involving decision-making, eventually based on plans, and action execution and monitoring. When speaking about joint action, we think it is the component that should handle coordination, communication, monitoring, repair strategies and eventually joint attention and common ground alignment, based on shared representations.

We can find in the literature multiple works proposing components with a part of these features. The ones that we will present have been a source of inspiration, from far or close, for the contributions of this manuscript. We will start with the oldest one, Shary, which has been developed in our laboratory. It is a component dedicated to supervision for human-robot interactions, with a strong emphasis on communication, allowing to execute shared plans and to monitor human and robot actions [75]. Chaski is a task-level executor, focusing on coordination and decision-making. It takes as input shared plans with deadlines and minimize the human idle time when executing of this plans [264]. There is also Pike an online executive that unifies intention recognition and plan adaptation to deal with temporal uncertainties during Shared Plan execution [151]. Görür and colleagues developed a robot able to handle unexpected human behavior, the first one being the human doing an action irrelevant to the task and the second one being the human not wanted the

robot assistance [119, 113]. For this, they developed a human model and have a monitoring of human's actions and endow the robot with the abilities to be reactive and proactive. Similarly, Baraglia *et al.* proposed a reactive and proactive robot, being able to help when requested by the human or when detected [15]. Iocchi *et al.* presented a framework which generate and execute robust plans for service robots [140]. It allows to not explicitly represent all possible situations would face (*e.g.*, low battery means the robot should not navigate) and also to face unpredicted situations where an action failed with no alternative solutions. They implemented it by separating the state variables needed at both planning and execution and the one needed at execution time only. Finally, Devin *et al.* implemented a supervisor allowing the robot to estimate the human's mental state about the environment and the states of the goals, plans and actions, while executing shared plans [87].

4.2 The Needs and Wants of a supervision system to manage interaction

A part of the control features presented here is inspired from Sandra Devin [86]. Indeed, we intended to pursue her work, re-implementing a part of her software using Jason, a BDI framework presented in Section 4.3, instead of if/else statements in C++, giving our software more flexibility, readability and genericity. Then, to go further, we developed JAHRVIS, a more complete approach of a supervision component dedicated to HRI which tries to satisfy multiple requirements:

- **Be generic**. The objectives developed in the rest of list are to reach for most collaborative tasks. Thus, it seemed essential for us to develop a software not dedicated to a particular human-robot task but able to handle plans for varied tasks.
- **Take into account the human partner**. In HRI, the human and the robot are partners. As seen in Section 1.3, partners perform better when taking each other into account. Thus, by considering human abilities, perspective and mental states, the supervisor makes the robot a better partner for the human.
- **Leave decisions to the human**. In some cases, it is not useful, even counterproductive that the robot plans everything beforehand. Indeed, such elements such as the human action parameters, or who should execute a given action when it does not matter, or the order in which some actions should be executed, can be decided at execution time. Thus, we propose a supervisor handling two types of plan allowing to give latitude to human decisions and actions: conditional plans, and plans extending "Agent X" shared plans [88].
- **Monitor human actions**. To monitor the plan progress, the robot should be able to monitor the human, *i.e.*, recognize their actions or be able to tell if they are idle.
- **Handle contingencies**. The robot has a shared plan, this is one thing, but to execute it and lead to the goal success is another one. Indeed, first, it

is not sure that the human has exactly the same, and failures can happen (see Section 1.5). Therefore, sometimes not everything is like the robot had planned and the decision and execution manager has to tackle this. Thus, it should be able to handle a certain amount of contingencies.

- **Manage relevant communications**. As stated in Section 1.4, communication is one of the key of collaboration. Therefore, it is important to endow the robot with the ability to manage relevant communication actions, verbal and non-verbal.
- **Consider the interaction outside collaborative tasks** A robot dedicated to collaborative tasks, in a real-life context, will interact with humans outside or between these tasks. We propose to consider this fact by defining what we called *interaction sessions*. An interaction session gives a frame to the interaction and allows to take into account a number of facts from one task to another or from one session to another.
- **Adapt to the human experience, abilities or preferences**. Humans are all different, because of their experience, abilities or preferences among other things. A robot taking into account its previous interactions with a human (*e.g.*, behaving differently with a novel user or an experienced user) or adapting to their abilities (*e.g.*, some people cannot climb stairs, a robot guide can indicate the elevator instead) will improve the efficiency and the quality of the interaction, and the user's experience.

4.3 Which tool to implement supervision?

4.3.1 The Choice of the Programming Framework

Restart from scratch or base oneself work on an existing software? This is the question which has been studied at the beginning of this thesis work about the implementation of the supervision software. It was possible (1) to develop the wanted features using the code[1] of the previous PhD student working on the supervision, Sandra Devin, (2) to choose among existing software dedicated to decision and execution for Human-Robot Interaction, (3) to choose among existing software dedicated to decision and execution for robotic platforms, and (4) to develop a new software from scratch.

The obvious drawback of (4) is that it takes a lot of time to start a new software from scratch and that it often leads to reinvent the wheel. Then, first we looked at existing solutions. Concerning the possibility (1), Devin had developed interesting features but the code is not modular and it was difficult to add new features or to modify the existing ones without breaking everything. Thus, there was the solutions (2) and (3) left. When looking for existing software to manage human-robot interactions, we could not find any open-source one with a minimum of features, documentation and not entirely dedicated to a given task. Therefore, we turned

[1] `https://github.com/laas/supervisor`

ourselves toward robotic frameworks. We compared existing open-source decision-making and execution software for robots. To cite a few, there is the PetriNetPlans library introduced by Ziparo *et al.* [309] which is a framework for planning and execution. Beetz *et al.* developed CRAM, a software implementing reasoning mechanisms that can infer control decisions [25]. A framework to implement hierarchical state machines is available among ROS libraries, SMACH[2], defined as "task-level architecture for rapidly creating complex robot behavior". Or, a C++ library to create behavior trees has been developed, called BehaviorTree.CPP [3]. Finally, there are several implementations of the BDI model presented in Section 3.1 such as JAM [135], Jadex [44], SPARK [205], dMARS [89], OpenPRS [139] or Jason [35].

As a first step, for prototyping and respecting project deadlines, our choice went to SMACH because its compatibility with ROS and its facility to be used. Then, it was no surprise, it became more and more difficult to program complex robot behaviors, state machine were not enough powerful. Thus, we examined possibilities for our second choice. After a comparison considering potential compatibility with ROS, possible integration with the other software of our architecture, availability of documentation, users' feedbacks, maintenance, and possibility of code modifications, our choice went to Jason designed by Bordini *et al.* [35] which is a Java interpreter of AgentSpeak created by Rao [227]. It has the advantage to be a BDI (Beliefs, Desires, Intentions, see Section 3.1) agent-oriented framework, fitting with our architecture. BDI frameworks implement a process, called the reasoning cycle or more commonly the sense-decide-act cycle [4], deciding step by step, which action to perform to reach a goal. It allows more modularity than state machines to handle contingencies and events. It also facilitates reasoning on agents' – humans and robot – beliefs. We chose this framework among the BDI ones and not another because it is implemented in Java and thus was compatible with rosjava[4] (*i.e.*, ROS implementation in Java), it is still developed and maintained, it is well documented (theoretically [35] and implement-ally[5]) which allows source code understanding and modifications, and there is a mailing list for users and its archives available[6].

4.3.2 Programming with Jason

As said above, Jason is a BDI-based framework, allowing what is called *agent-oriented programming*. Originally designed for multi-robot programming, it can be used for other purposes such as ours. How does it work?

We explained in Section 3.1 that there were three main concepts involved in BDI models: beliefs, desires and intentions. Well, Jason's purpose is to program agents. Thus, each agent has beliefs, desires and intentions. The beliefs are what it perceives, acquires from other agents and computes. They can produce desires, *i.e.*, states of affairs the agent wants to achieve. Then, the agent deliberates on

[2] `http://wiki.ros.org/smach`
[3] `https://github.com/BehaviorTree/BehaviorTree.CPP/`
[4] `https://github.com/rosjava/rosjava_core`
[5] `http://jason.sourceforge.net/api/`
[6] `https://sourceforge.net/p/jason/mailman/jason-users/`

its desires and choose to commit to some of them, *i.e.*, the chosen desires become intentions. To satisfy its intentions, the agent executes procedural programs, called plans, leading to actions. The procedural knowledge is written by the programmer.

The programming of the behavior of an agent is in the AgentSpeakLanguage (ASL). The program is designed by a user, a programmer. A program contains, among other things, plans. These plans have actions. An action is described by a Java program, written by the Jason's user. Then, to run, a program uses the decision loop, so called the *reasoning cycle*, integrated to Jason. It is possible to customize some functions of the reasoning cycle by overloading or adding Java functions of the agent's constructors, belief base and reasoning cycle.

4.3.2.1 Agents

In the ASL program of an agent, it is possible to see plans, beliefs, desire and test goal. First, let's see a very simple example of program with the agent Bob[7], presented in Listing 4.1. Bob has one initial (*i.e.*, given by the programmer, not acquired by perception) belief which is `happy(bob)`. A belief is a property, here `happy`, which can have whatever number of arguments (including zero), here `bob` and a source (*e.g.*, `source(percept)` means that the belief has been acquired through perception, `source(self)` means that it has been computed by the agent itself and `source(alice)` means that it has been received from the agent Alice). Then, he has one initial desire which is recognizable by `!`. And finally, he has a plan allowing to achieve the desire `say(hello)`. A plan is triggered by an event, here `+!say(X)` (*i.e.*, the event is that the goal `say(hello)` has been added), has a context (*i.e.*, a precondition), here `happy(bob)` and has a body which contains the actions to execute, here `.print(X)` (with `X` being a variable – variables have their first letter in upper case). If we remove the initial belief `happy(bob)` from the first line, as the program is written and considering that Bob is the only agent, he cannot print hello, as the precondition of the plan will not be true.

```
happy(bob). \\belief

!say(hello).  \\desire

\\plan
+!say(X) : happy(bob) <-
  .print(X).
```

Listing 4.1: ASL program of Bob, a Jason agent

In another example, illustrated by Listing 4.2, Bob has no initial belief nor initial goal. He has plans for two events: starting to believe he is happy and having the goal to say hello. We can see that there is also a program for another agent, Alice.

[7] `http://jason.sourceforge.net/mini-tutorial/hello-bdi/`

She has an initial goal, her, which is to inform bob that he is happy. Therefore, we can see that an agent can add a belief in another agent's belief base. When Bob gets the information that he is happy, this triggers his first plan, creating for him the goal `!say(hello)`. As Bob does not believe that today is Monday, he can trigger his second plan to say hello. In this plan, there a three elements: a print action, a wait action and the addition of a new goal. And thus, here, we are in the presence of a recursive plan which never ends.

```
\\bob.asl
\\for example purposes, the precondition is true
\\but it can be logical expressions with beliefs,
\\functions...
+happy(bob) : true <-
  !say(hello).

+!say(X) : not today(monday) <-
  .print(X);
  .wait(500);
  !say(X).

\\alice.asl
!inform.

+!inform : true <- .send(bob,tell,happy(bob)).
```

Listing 4.2: ASL programs of Bob and Alice, two Jason agent

4.3.2.2 Actions

To give an idea of what looks like the Java program of an action, here is an example of a Java function for the action `.print` in Listing 4.3.

```
public class print extends DefaultInternalAction {
  @Override
  public Object execute(TransitionSystem ts, Unifier un,
     Term[] args) throws Exception {
    String sout = argsToString(args);
    System.out.print(sout.toString() + "\n");
  }
  return true;
  }
}
```

Listing 4.3: .print action

In Jason, there are two types of actions defined: *environment actions* and *internal actions*. *Environment actions* allow an agent to act within its environment, usually producing effects visible by other agents. Whereas, *internal actions* are designed to be run internally within an agent such as the print action and can be used to return values or booleans. When being executed, there are not handled the same way in the Jason's reasoning cycle. The definition of which type an action should be falls to the programmer, which should choose according to their need.

We have seen what looks like the program of Jason agent. Now, we are going to see how it is run by the Jason interpreter.

4.3.2.3 Reasoning cycle

Each agent has what has been coined a *reasoning cycle*, composed of 10 steps. It resembles a decision loop, running each step one by one and starting again at the first one. The steps 1 to 4 are dedicated to the belief update of the agent. The steps 5 to 10 describe the interpretation of the ASL program. In these latter, an event is selected, as well as a plan corresponding to this event and then the first formula (*e.g.*, an action or a goal) of the plan is executed. It is illustrated by Figure 4.1. The steps are the following ones, in this order:

1. Perceiving the Environment: Each agent has a Java function called `perceive`. This function can retrieve data from a simulated environment or be customized by the programmer to get actual perception data. The function outputs a list of beliefs, along with their source (*e.g.*, `<isOn(box1,table)[source(percept)]`, `color(box1,red)[source(percept)]>`).

2. Updating the Belief Base: The agent's belief base is updated with the perception data. Each change in the belief base generates an event (*e.g.*, `+color(box1,red)[source(percept)]` and if later the color of the box is not part of the perception data anymore, it will be `-color(box1,red)[source(percept)]`).

3. Receiving Communication from Other Agents: It checks if an agent received a message from another agent such as the message Bob received from Alice in Listing 4.2. A message can be a belief, a plan, a goal or a questioning on a given belief.

4. Selecting 'Socially Acceptable' Messages: It is a function the programmer should customize. It allows agent to refuse messages or types of message from some given agents based on some rules written in Java by the programmer, *e.g.*, no message from the agent Alice.

5. Selecting an Event: Events are either perceived changes in the environment or changes in the agent's own goals. There is a queue of events and at each reasoning cycle only one is selected to be handled. The default method to

select it is a FIFO but, as every function of the reasoning cycle, it can be customized.

6. Retrieving all Relevant Plans: From the selected event, it tries to find all the relevant plans for this event, in the plan library, *i.e.*, the plans written by the programmer in ASL. The function tries to find the plans that can be *unified* with event, *i.e.*, the ones with their left part (the trigger) matching the event. For example, if the selected event is `+color(box1,red)[source(percept)]` and in the plan library there are these six plans:

```
+position(Object,Coords) : true <- .print(Coords).
+color(Object,red) : true <- .print(nice).
+color(Object,red)[source(self)] : true <- .print(
   nice).
+color(box1,Color) : true <- .print(nice).
+color(Object,Color) : false <- .print(Color).
+color(Object,blue) : true <- .print(so-so).
```

then there are three relevant plans (the last one is also relevant because what is looked for here is the triggers only and not the preconditions):

```
+color(Object,red) : true <- .print(nice).
+color(box1,Color) : true <- .print(nice).
+color(Object,Colour) : false <- .print(Colour).
```

7. Determining the Applicable Plans: It takes the list of relevant plans and sees which ones are applicable. To do so, it looks at the context (the preconditions) of the plans. The context can be beliefs, prolog-like rules, internal actions, logical expressions or booleans. If we look at the example of the previous step, there were three relevant plans. Their contexts are simple booleans. Two of them are true, the other one is false, thus the two first plans are applicable.

8. Selecting One Applicable Plan: It takes the list of applicable plans and selects the one that will be elected to become an intention, *i.e.*, to be executed. As usual, this is a customizable function for which the default behavior is to take the first plan in the order of the plan library, *i.e.*, in the order written by the programmer. Still with the same example, thus, the one plan to be selected is the first one, `+color(Object,red) : true <- .print(nice).` If the event was external, *i.e.*, from perception, it creates a new intention, adding it to the set of intentions. Then, the agent has a new *focus of attention*. If the event was internal, *e.g.*, a belief addition inside a plan, then the selected plan is added on the top of the existing intention.

9. Selecting an Intention for Further Execution: As seen in the previous step, an agent can have more than one intention in the set of intentions, each representing a different focus of attention. Then, at this step is chosen the

intention of which the formula will be executed. The default function chooses the first intention of the list. After execution of the formula, the intention will go at the end of the intentions list.

10. Executing One Step of an Intention: The first formula of the selected intention is executed (this number is also customizable and the programmer can choose that an agent execute more than once formula in the same reasoning cycle). It can be an internal action, an environment action, a goal, a belief addition or deletion and two other types that will not be developed here.

Therefore, each agent has a reasoning cycle running repeatedly, independent from the other agents' reasoning cycle. Interactions between each agents happen through the messages they send to each others and eventually the effects they produce on the environment which are then perceived by the other agents.

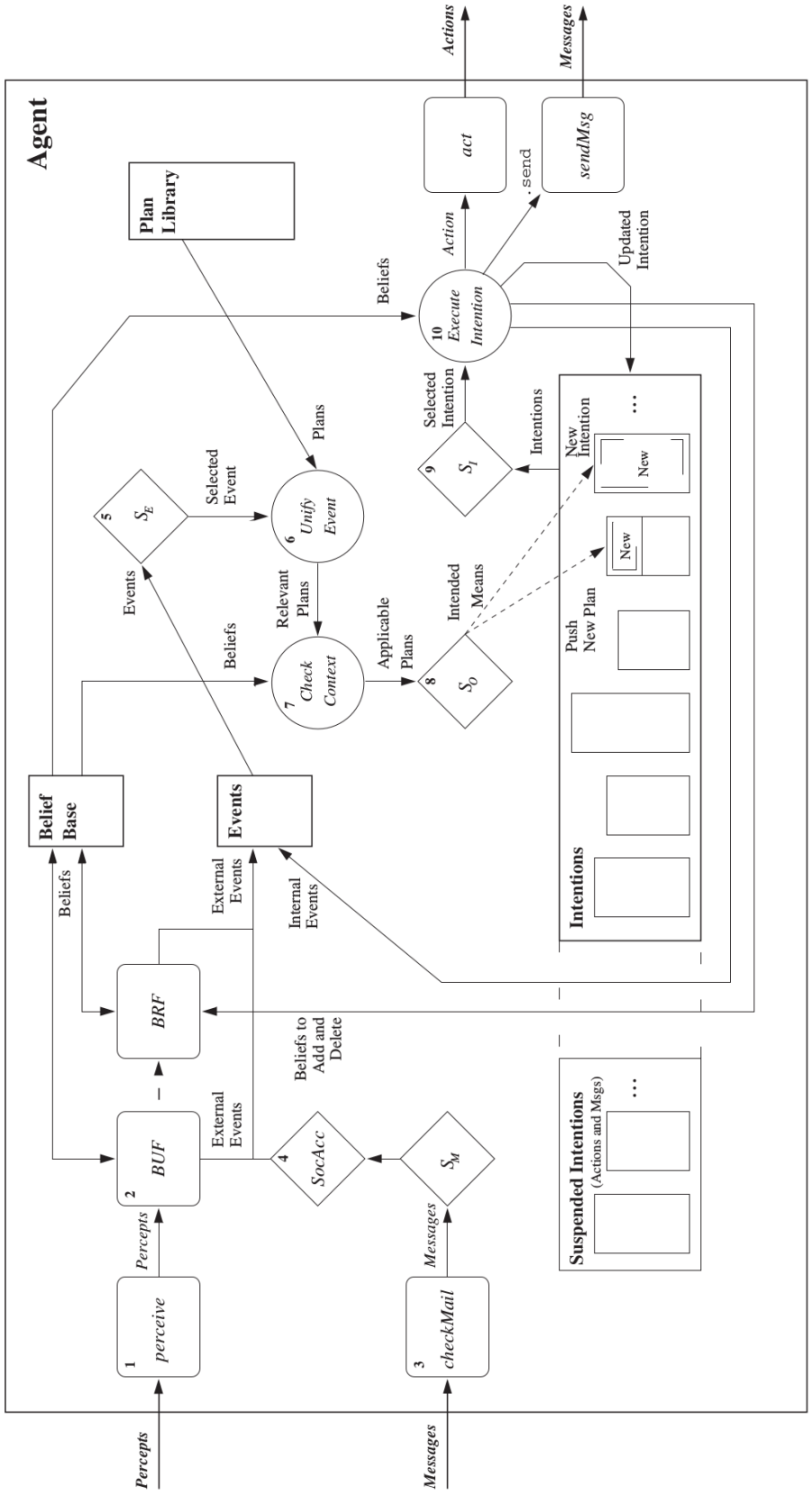


Figure 4.1: The Jason reasoning cycle [35]. Each step presented above has its numbered corresponding box.

4.3.2.4 Plan failure handling

For the case where a plan fails (*e.g.*, an action fails – there are other reasons for which a plan could fail but we will not discuss the details here), Jason integrates a mechanism handling failures. It consists in cancel the execution of the plan and generating a triggering event for a contingency plan whose prefix is `-!`. If the contingency plan can be found – written by the programmer –, it is executed. Then, if the plan which originally failed was a subplan of another plan, this plan will continue normally.

An illustration is given in Listing 4.4. The result of the execution of this agent file would be the printing "unknown error" and then "bye" in case of the failure of the `robot_speech` action execution with an instantiated speech module. Indeed, the initial goal `speak` creates the subgoal `say_hello`. Unfortunately, the action `robot_speech` fails with an empty error message, generating the event `-!say_hello[error_msg(Msg)]`. There are two plans for this event but as `Msg=""`, the second one is chosen, printing "unknown error". Then, `speak` continues in the same way it does when goal `say_hello` is achieved successfully, printing "bye".

```
!speak.

+!speak : true <-
!say_hello;
.print(bye).

+!say_hello : true <-
robot_speech(hello);
.print(hello).

-!say_hello[error_msg(Msg)] : .substring(Msg,
   no_speech_found) <-
.print(no speech module was found).

-!say_hello[error_msg(Msg)] : true <-
.print(unknown error).
```

Listing 4.4: Example of plan failure handling

4.3.3 Jason Integration with ROS

The robotic architecture presented in Section 3.3 uses the ROS framework [225] to enable communication between its components. Thus, to be able to build a supervision software based on Jason, we needed to interface it with ROS as well. At the time, there was no available bridge between Jason and ROS, Jason being extensively used in simulation contexts. Thus, we developed our own – and at about

the same moment, the Jason's developers started to develop theirs [268] (what we realized a bit later), both using rosjava. We tackled the problem in very different ways. A user of their implementation only needs to fill one perception (topics) and one action (topics/services) manifests to link the system with ROS and then implement their agent in ASL. Thus, it is quite easy to use. However, it has drawbacks. Therefore, action requests are directly sent from ASL to the hardware controller, with no possibility of Java processing. Moreover, action status/result can only be boolean which is not enough for a system like ours needing to perform service queries of data to the external Knowledge Base for example. Finally, there is no bridge with action servers which are often used for motion planers for example.

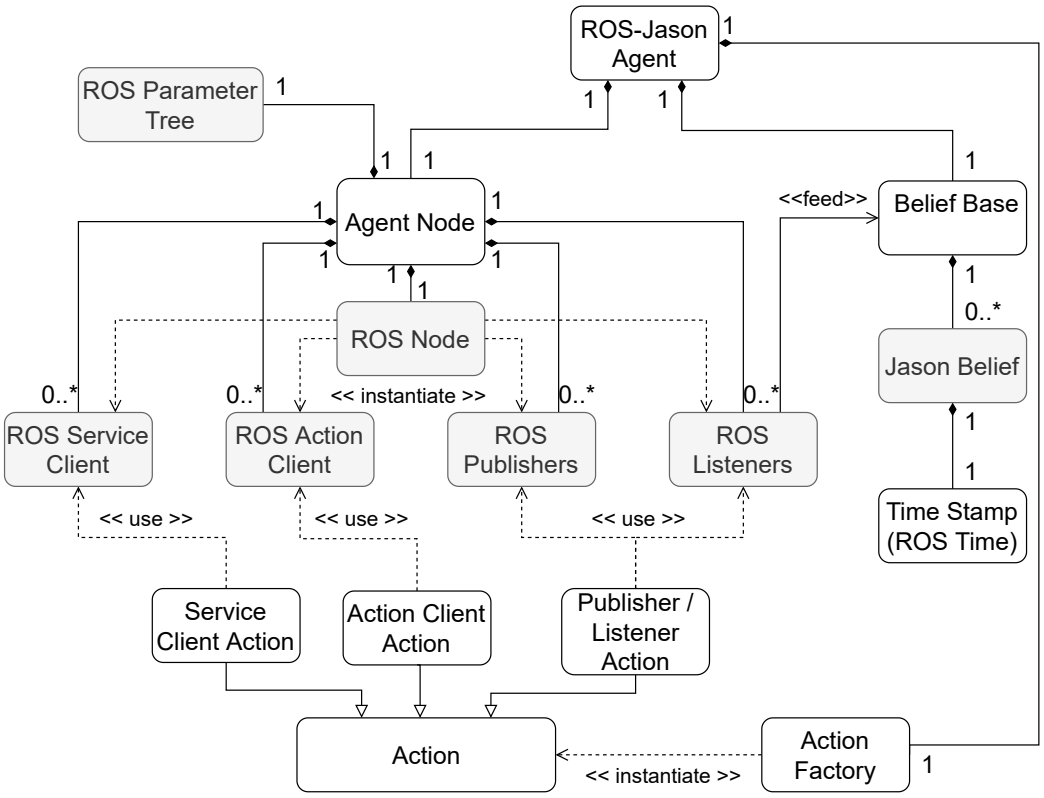


Figure 4.2: Simplified Java class diagram of our ROS-Jason implementation. In white are our customized classes and in grey the native ROS and Jason classes.

A simplified Java class diagram of our implementation[8] is presented in Figure 4.2. We defined for each Jason agent a customized Java class (ROS-Jason Agent (RJA) on the figure) which has an Agent Node, an action factory and a belief base where all beliefs are time stamped with the roscore time.

An Agent Node has an attribute, the ROS Parameter Tree, allowing to load YAML parameters from files in which, among other things, are written services, topics and action servers info, as shown in Listing 4.5, a bit similarly to the manifests

[8] `https://github.com/amdia/rjs`

of [268]. From these parameters, the Agent Node can automatically instantiate all the needed ROS components through its ROS node.

The Belief Base can receive perception updates through ROS topic listeners. Moreover, we customized[9] the belief update function (step 2 of the reasoning cycle) as we chose to abandon a state-based perception to adopt an event-based perception. So, percepts are not elements that when perceived at time T are added to the belief base and disappearing when not perceived anymore at time T+1. There are now updates (additions and deletions) from the external Knowledge Base, in this way, it limits the number of message exchanges, *i.e.*, instead of receiving every 500 ms during 10 seconds that the agent perceives `cube_1`, it receives for example an addition at t=18s and a deletion at t=28s.

To each belief added in the belief base, from perception or internal computation, is added a time stamp from the current ROS time. Currently, it is useful for the computation of the Quality of Interaction presented in Chapter 7, to feed the KB timeline and for debugging.

An RJA has an Action Factory – abstract in the ROS-Jason framework and instantiated in JAHRVIS – containing the list of environment actions it can perform – in the case of our architecture, not all actions of this type are for the robot to act on its environment, sometimes there are queries to other components of the architecture. The Action Factory instantiates the Action called through the ASL program at execution time. An Action can either be based on a ROS service client, or an ROS action client, or a ROS publisher for the request and a ROS listener for the result.

```
services:
  onto_individual:
    name: /ontologenius/individual/robot
    type: ontologenius/OntologeniusService
  onto_class:
    name: /ontologenius/class/robot
    type: ontologenius/OntologeniusService
topics:
  mementar_occasions:
    name: /mementar/occasions/robot
    type: mementar/MementarOccasion
    function: sub
  plan_request:
    name: /planner/request_new_plan
    type: planner_msgs/PlanRequest
    function: pub
action_servers:
  plan_motion: /pr2_tasks_node/plan
```

[9]This modification of the belief update function is not part of our ROS-Jason implementation but is on top of it, in the JAHRVIS implementation which relies on ROS-Jason.

```
  execute_motion: /pr2_tasks_node/execute
```

Listing 4.5: Example of service, topic and action server definitions in a YAML file.

Part III

Joint Action-based Human-Aware supeRVISor: JAHRVIS

Chapter 5

JAHRVIS by the menu

Contents

In the previous chapter, we presented all the previous works we got our inspiration from, from psychology to robotics by way of philosophy, sociology and neuroscience. What is a social interaction? how can it be divided in steps? what is a joint action? how humans collaborate together? how do they take into account their partners? what happens when an agent makes a mistake? what has been done in computer science or robotics until now to make robots better collaborators? All these theories, ideas, questionings nourished our thoughts for the design and implementation of a supervision system dedicated to Human-Robot Joint Action. Supervision is key in the architecture as it is the robot decision kernel. And, as most components of a robotic architecture dedicated to HRI, one of the main issues of supervision is how to take the human into account, a more or less unpredictable agent with whom the robot has to collaborate.

We presented in Section 4.1 a few works tackling supervision issues, *i.e.*, how to adapt to the human, how to monitor them, how to face unexpected human behavior, how to optimize the task efficiency, how to make the robot an good human helper... They were very inspiring but we found out it was missing a general architecture and a software that could be used in different types of collaborative tasks, available for the community and that could easily be enhanced with new features. These thoughts led to the development of the Joint Action-based Human-aware supeRVISor (JAHRVIS) which is the central topic of this chapter. We also came up with a novel idea: to endow the robot with the ability to measure if an interaction is going well or not. Such ability can be used by the supervision to enhance its adaptation capacity.

In the two first sections, we present the role and features we defined for JAHRVIS. Next, in Section 5.2, we present our representation of Human-Robot collaborative activity. Finally, we introduce JAHRVIS overall structure in Section 5.3 whose role is to decide and control the robot during an interaction.

5.1 The Role and Features of JAHRVIS

JAHRVIS is a supervision system, *i.e.*, it embeds the robot high-level decisions, controls its behavior and tries to react to contingencies, whenever necessary considering the human the robot is interacting with. Thus, JAHRVIS is to differentiate from supervision systems dedicated to robot or multi-robots control as humans are taken into account.

It queries, manages and executes (shared) plans which are (partially) ordered set of actions to be performed by human and robot agents in order to achieve a (shared) goal. The plan management, described in Section 6.4, is based on the estimation of the human's mental states, its knowledge about the current state of the environment, and recognized human actions. We explored the management of various kind of plans: (1) shared plans in which each action is allocated to an agent as well as action parameters are given objects, (2) shared plans in which actions might not be allocated to an agent at planning time and parameters might refer to objects with a semantic query, and (3) conditional plans which anticipate different possibilities for the human decision/action.

As mentioned previously, the plan management relies on the recognition of human actions, among other things. JAHRVIS integrates its own processes of action monitoring, *i.e.*, selecting the robot's point of interest and enabling joint attention, presented in Sections 6.4.1 and 6.5, and of action recognition. This latter process, introduced in Section 6.3, is model-based and have been designed to be robust to a potentially unreliable perception of the human.

As there are actions of the plan to execute by the robot, JAHRVIS needs interfaces with the robot controllers. Moreover, actions can be of two types, physical and communicative actions, and so requires a differentiated management. The methods implemented to handle the action execution will be introduced in Section 6.5.

Finally, an important feature is the ability to verbally communicate with the human. Indeed, during a collaborative task, communicate might be needed, among other things, to inform the partner of a performed action, or to ask them to perform one. Section 6.6 describes the choices we made to endow the robot with a minimum set of communication abilities.

5.2 Representation of a Human-Robot collaborative activity

It is possible to describe and decompose a Human-Robot collaborative/joint activity in various ways for (see Section 1.3.1 for discussions related to joint or collaborative activities). What we define as collaborative activities or tasks are types of joint actions. For all the following definitions, we place ourselves in the context of one-to-one human-robot interactions, however we believe that the scheme can be extended to multi-human multi-robot contexts. We draw our inspiration from the literature of sociology and robotics, presented in Section 1.1 and Section 2.1,

to define a model of interaction with three layered levels: interaction session, tasks and actions; as illustrated in Fig. 5.1. We chose to represent collaborative tasks and their decomposition using the Hierarchical Task Network (HTN) [104] representation which is often used in cognitive robotics [138, 176, 53] and because it allows to deal with goal-based and situation-based activities at different levels of hierarchy such as task, subtasks and actions and consequently to consider different levels of granularity.

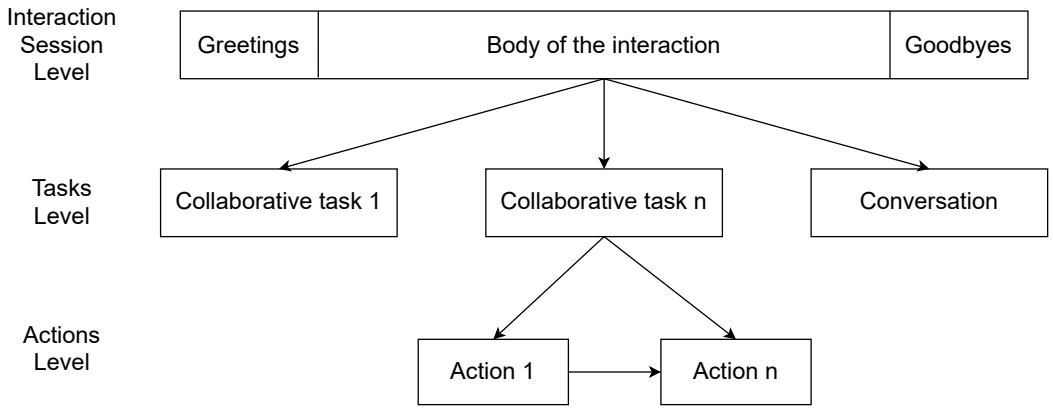


Figure 5.1: The hierarchical structure of an interaction session. The highest level is the interaction session. The second level is composed of the tasks. They are included in the body of interaction of the interaction session and, two types of tasks are considered and may overlap, collaborative and conversational tasks. With this representation, a task can be recursively refined as subtasks until reaching the last level, the actions level, which is considered as atomic.

5.2.1 Representation of a Human-Robot Interaction Session

We define an **interaction session** as the period during which the robot and a human interact together and are engaged. It is divided in three parts, following the structure proposed by Robinson [234] as presented in Section 2.1: the greetings, the body of the interaction and the goodbyes. First, *the greetings* corresponds to the period where an agent starts an interaction by initiating it with another agent. The interaction session lasts as long as the interactants are maintaining the interaction through conversation and collaborative tasks performance which corresponds to the *body of interaction*. Finally it ends when at least one of the interactants is disengaged, either by abruptly ending the interaction or by closing the interaction as described by Schegloff and Sacks [255], it corresponds to "the goodbyes". For example, for an entertainment robot in a mall, an *interaction session* starts when a person signals to the robot that they want to engaged, by greeting it or by approaching it and looking at it. The body of interaction is composed of conversation and eventually direction-giving tasks and, the session lasts until the person says goodbye or leaves. This is the nominal case and, the duty of the robot

is to contribute to maintain the session alive until the human decides to close it, because it is at the service of humans. However, in some (extreme) cases, the robot might decide to close the interaction by itself.

Moreover, as seen in Section 1.3.3.2, social interactions and joint activities (or actions) involve commitment, or rather engagement as we say in robotics – this difference in the vocabulary as been highlighted in [62]. As explained in the previous chapter, there is no unique definition of what it means to be engaged. We chose one that is frequently used in robotics, proposed by Sidner and Lee [266]: "Engagement is the process by which two (or more) participants establish, maintain and end their perceived connection during interactions they jointly undertake". The robot must be able to exhibit its engagement and disengagement and also to assess them with respect to its human partner.

We defined three states for the body of interaction, corresponding to what can happen during the latter:

- conversation: a social chit-chat or a goal negotiation, without any physical action performed except communicative gestures
- collaborative task: both agents executing actions in order to achieve a shared goal
- idle phases: the agents are not chatting or performing a collaborative task together but remain engaged in the interaction session, it happens in-between active interaction phases

For each of these three states, the way to exhibit the engagement varies (*e.g.*, in a conversation, an agent looking at their partner displays their engagement; during a task, an agent correctly performing their action is a way to demonstrate their engagement). That is why there is a need to define what behavior the robot has to exhibit in each state and what behavior it should expect from the human in each state, as these behaviors are usually very specific (*e.g.*, in a direction-giving task, the robot keeps its head oriented toward its partner's face to demonstrate its engagement in conversation and idle contexts and when it gives a direction it expects the human to look at the direction it is showing; in a stack task, when the robot gives an instruction it expects the human to take a given cube).

Transitions from one state to another can be managed by triggers more or less complex. For example, a collaborative task can be initiated when a human asks the robot that they achieve a goal together.

5.2.2 Collaborative Tasks, Substasks and Actions

Tasks compose the body of the interaction of an interaction session as shown in Fig. 5.1. We distinguish conversation (*i.e.*, agents engage in dialogue to exchange ideas, or to ask questions) from collaborative tasks (*i.e.*, agents work as partners, collaborating to perform tasks and to achieve common goals). We will not develop more on conversation since it is not the main focus of this work.

In joint or collaborative activities (see Section 1.3.1), humans are committed to achieve a goal together, involving collaboration and shared plans as shown in

Section 1.3. And, when interaction with a robot, the same mechanisms are also triggered as they are essential to a successful collaboration. When a human and a robot perform a task together, as described by Bauer *et al.* [22], we could say that the robot has the intent to help the human, so the human's intention becomes its own intention. Then, they have the joint intention to reach a common goal and, as shown by Michael and Salice [198], they have a commitment to the joint activity, leading to perform joint actions. Therefore, during its evaluation and decision-making processes, the robot has to take into account that the human and itself should remain engaged all along an interaction session for the tasks to be successful and both have to manage and contribute to maintain expectations about what the other is doing.

The elements composing a *task* are: a goal, a plan and involved agents. A plan is needed to achieve a goal. The ones we manipulate are HTN-based plans, composed of *abstract tasks* that we also call *subtasks* and *primitive tasks* that we also call *actions*.

Actions are the elementary items of tasks, *primitive tasks*, manipulated by the high-level robot supervision controller. They cannot be decomposed further by it (*e.g.*, placement and motion planning are achieved by a lower control system not described here). It is usual to describe an action with its preconditions, its effects and, the agents and entities implied in its execution (*e.g.*, in plans written in PDDL (Planning Domain Definition Language) [103]). We add to this description the notion of expected reactions (which can themselves be actions) from the other agents once the action is executed.

In our model, an agent (human or robot) is a contributor to the task and has a mental state as described by Devin *et al.* [87]. The mental state is a set of facts representing, from the agent point of view, the current world state, the state of the goal and the current task state. Since we are interested here in the robot situation assessment and decisional processes, the mental state of the human is built and managed by the robot as an estimation of the beliefs of the human [201, 128, 280].

5.3 The Structure of JAHRVIS

The Joint Action-based Human-aware supeRVISor (JAHRVIS) is implemented on top of our ROS-Jason framework[1]. During the design of JAHRVIS, we identified seven high-level features we needed and implemented their associated processes, based on the objectives presented in Section 5.1[2]. We present JAHRVIS structure in Figure 5.2, with the seven processes in blue, the QoI Evaluator dedicated to the interaction evaluation and the six others to the decision and control. All the next developments of this chapter will be about the description of these processes. The components not in blue are external components from the robotic architecture

[1] `https://github.com/amdia/ld_rjs`

[2] The design of JAHRVIS was an iterative work, indeed the first version being the supervisor implemented for the task described in Chapter 8, the second one was the supervisor of the task described in Chapter 9 and the final one was the supervisor of the example used all along Chapter 6

presented in Section 3.3 to which a part of knowledge maintenance, decision-making and execution are delegated by JAHRVIS.

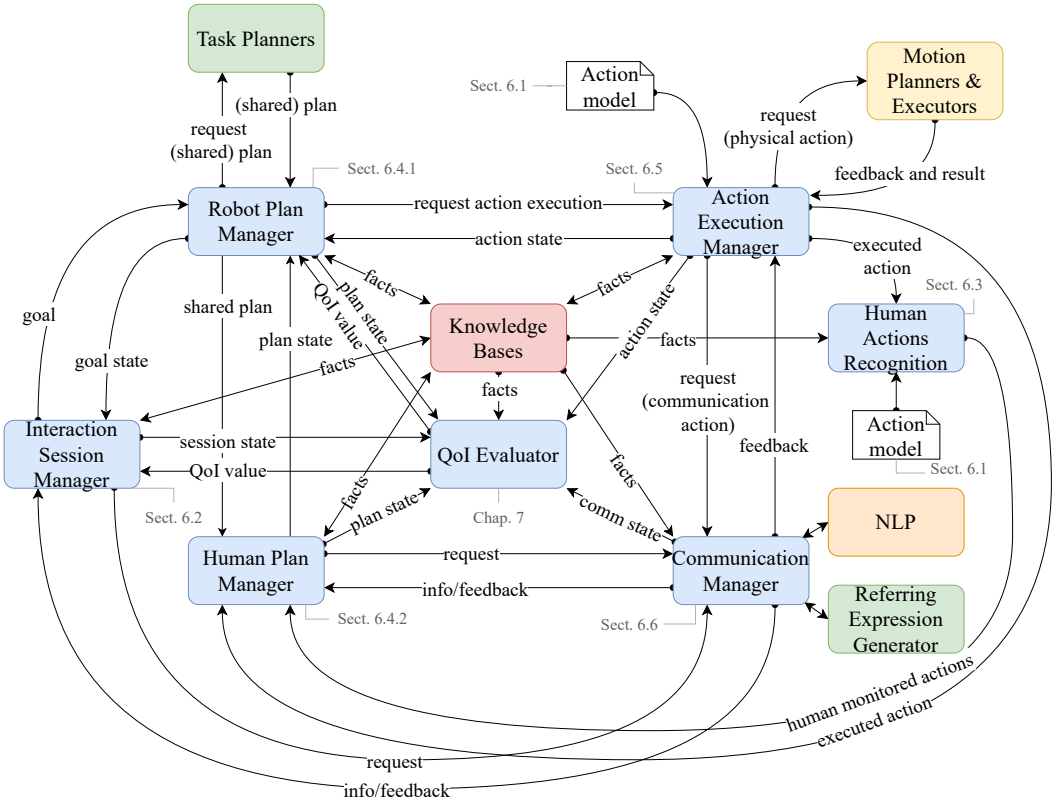


Figure 5.2: The JAHRVIS processes (in blue) and their interactions between themselves and with the other components of the robotic architecture presented in Section 3.3.

update section numbers

For each process (in blue in Figure 5.2), we implemented a ROS-Jason Agent (RJA), with the desired behavior coded in ASL and the needed customizations added in Java. Thus, internal communications between the JAHRVIS processes, and so RJAs, use Jason messages (see Section 4.3.2). External communication with the other components of the robotic architecture is based on either ROS messages, services or action clients.

Not all RJAs are active at each level of interaction defined in Section 5.2. Indeed, as its name suggests, the *Interaction Session Manager* handles interaction sessions. The *Robot* and *Human Plan Managers* handle the task level. And, the *Action Execution Manager* and the *Human Actions Recognition* are in charge of the action level. The *Communication Manager* is active at all levels. We can also make the distinction between the component dedicated to the assessment of the quality of interaction, *i.e.*, the QoI Evaluator which will be described in Chapter 7, and all the other ones, dedicated to the decision-making and control.

Figure 5.3 shows an overview of the data representing the state of JAHRVIS at each instant when the system runs. The robot can either be in an interaction session with a human or be by itself. When it is in an interaction session, it computes the human's commitment (it may be a simple function checking if the human is here or not) and is available to perform collaborative tasks. When the robot is not interacting with humans, it can have tasks to perform such as going to its home base. If a collaborative task should start (on human request or on the robot's initiative), a (shared) plan is got from the Task Planner as shown in Figure 5.2. When the collaborative task is ongoing, the robot has its beliefs about the environment and the plan progress, and estimates the human's ones. Beliefs about the environment are provided by other components of the robotic architecture presented in Section 3.3: the Situation Assessment and the Knowledge Bases. Each abstract and primitive task has a number of data associated to it. Moreover, Quality of Interaction and human action recognition are continuously processed. Finally, when an action is executed by the Motion Planners and Executors, updates about the action states are communicated to JAHRVIS.

In the Chapters 6 and 7 will be presented these processes. Chapter 6 introduces the ones related to the decision-making and robot control while Chapter 7 describe the evaluation process of the Quality of Interaction. Chapter 6 will start by laying the foundations for the RJA functioning: the knowledge representations and management. Then, each RJA will be thoroughly described. The Interaction Session Manager (ISM) is dedicated to in-between tasks, *i.e.*, the opening and closing of interactions and all the dialog which can happen between two collaborative tasks. When a shared goal is established, the shared plan is handled by the Robot Plan Manager (RPM) and the Human Plan Manager (HPM), *i.e.*, to follow the plan progression, to make sure that the observed human actions match the ones of the plan and to decide when the robot should act. Robot actions to perform are sent to the Action Execution Manager (AEM) that interfaces with the motion planers and executors. As for human actions, they are monitored and recognized by the Human Actions Recognition (HAR). Finally, the Communication Manager (CM) is in charge of producing the communication for the human when requested by another RJA along with the human communication reception.

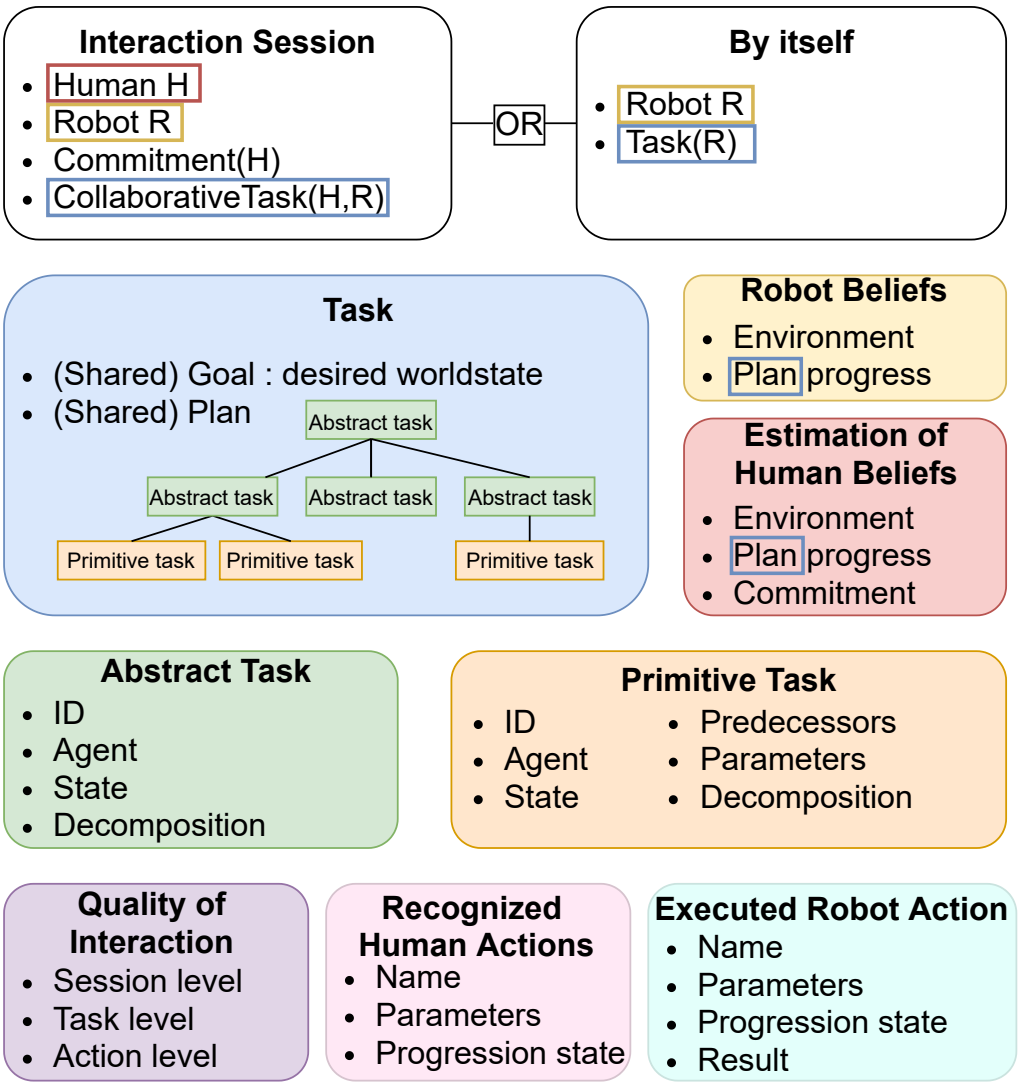


Figure 5.3: Overview of the data representing the state of JAHRVIS at each instant when the system runs. The robot can either be in an interaction session with a human or be by itself. When it is in a task, plan progress are maintained from the robot's and human's estimated point of views. A plan is composed of abstract and primitive tasks whose state evolved during the task. QoI and human action recognition are two elements continuously computed. When the robot executes an action, the state of the later is updated as well as its result (success or failure).

Chapter 6

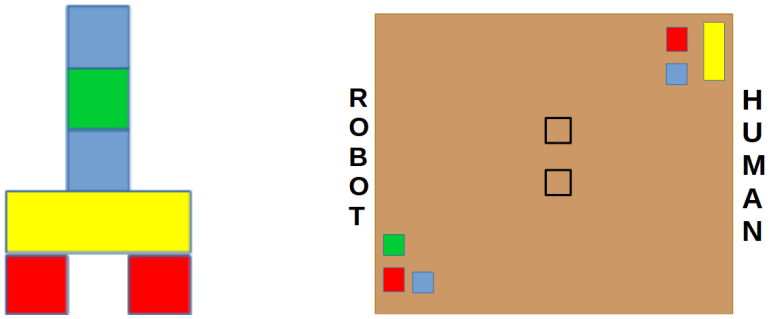
How JAHRVIS works

Contents

The objective of this chapter is to present the JAHRVIS processes involved in the decision-making and the control of the robot when jointly interacting with a human. First, we present the knowledge representations used, then the Interaction Session Manager, the Human Actions Recognition, the Shared Plans Handling composed of two processes, one for the robot (Robot Plan Manager) and one for the human (Human Plan Manager), the Action Execution Manager, and finally the Communication Manager.

Most examples given in this chapter will base themselves on a collaborative task, the *BuildingTask*, that we are going to present now. It has been inspired by [88]. A human and a robot have to build a block construction together as represented in Figure 6.1a. At the beginning of the task, the robot and the human have several colored cubes (the yellow one will be referred to as a stick) they can access as in a set-up like the one illustrated in Figure 6.1b. Two identical placements are set on the table to indicate where to put the two red cubes which are the first ones to place. Each agent has 3 available actions: Pick, Place and Wait. They can only access to the cube on their side of the table.

Figure 6.1 will be reminded in page headers where the BuildingTask example will be mentioned.



(a) Goal of the task (side view)

(b) Initial set-up (top view)

Figure 6.1: Description of the blocks building task. The human and the robot have to build the stack together. We assume that the robot and the human know where all the available blocks are. We would like the robot to adapt as much as possible to the human actions and decisions while avoiding useless or tiresome verbal interactions.

6.1 Knowledge Representations and Management

As shown in Section 1.3.3.4 and Section 1.3.3.6, when involved in joint actions, humans have shared representations of tasks, actions, goals and have a common ground. Thus, it is important that the robot has such internal representations.

During an interaction, JAHRVIS processes use knowledge bases (KB) of two types: internal and external ones. Concerning the first type, each RJA has its own knowledge base that we call belief base in Jason vocabulary. It is used for knowledge which serves to JAHRVIS internal computations only. As for the external knowledge bases, there are the ones presented in Section 3.3.3.2 as part of the robotic architecture, Ontologenius and Mementar. Updates from subscription to Ontologenius facts are received through ROS topics and converted as Jason percepts to be added to the subscribing RJA belief base.

Table 6.1 shows a description of the data circulating between the JAHRVIS RJAs and the 2 Knowledge Bases, Ontologenius and Mementar.

	RJA subscription to KB	RJA request/KB response	RJA feeding KB
Interaction Session Manager	*isEngagedWith*(Human, Robot) *isPerceiving*(Robot, Human) *isLookingAt*(Human, Robot)		Start and end of interaction sessions
Robot Plan Manager	*isLookingAt*(Human, Robot)	Class of actions SPARQL → Object	Start and end of abstract tasks
Human Plan Manager	*isPerceiving*(Human, Robot) *isPerceiving*(Robot, Human) *isLookingAt*(Human, Robot) *isLookingAt*(Human, Object)	Class of actions Existence of action effects SPARQL → Object	Start and end of primitive tasks
Communication Manager	*isPerceiving*(Robot, Human)	Verb conjugation Class of actions and objects Labels of actions and objects Existence of verbalization contexts	
Human Actions Recognition	Movements of the human action model Progression effects of the human action model Necessary effects of the human action model	Class of Objects Existence of preconditions Existence of isReachable(Object)	
Action Execution Manager		Class of actions	Start and end of primitive tasks

Table 6.1: Data circulating between the Knowledge Bases (Ontologenius and Mementar) and the JAHRVIS RJAs. Data in italics are not task-dependent, the other ones are. The types of the latter are loaded from the action models described in Paragraph: Internal Action Representation. For example, the Human Actions Recognition gets from the internal action model that *handMovingToward*(Human, Pickable) is a movement of the Pick action. Then, it can subscribe to the updates about this fact type to Ontologenius when a task where the human has a Pick action is ongoing.

6.1.1 Action Representations

Action representations allow

- the robot to recognize human actions
- to execute actions
- to monitor the human's attention towards its actions and to communicate about them

We defined three action representations according to these three uses. Actions should be written and loaded by the programmer according to the task they need the robot to perform, following the defined formats in the dedicated files. This allow JAHRVIS core to be task-independent.

Human actions to recognize and robot actions to execute are written in an ASL file to benefit from Jason reasoning features. And, the same robot actions but with other information allowing JAHRVIS to communicate about them and to monitor the human's attention towards them are stored in Ontologenius to benefit from the reasoning features. This latter representation is described in the next paragraph.

External Action Representation For the needs of JAHRVIS, we represented actions, their verbal labels and their effects in the semantic KB managed by Ontologenius. We show in Figure 6.2 a representation of some actions we stored in Ontologenius using the Web Ontology Language (OWL) (see Listing 6.1) and in Figure 6.3 a representation of possible action effects.

```
:PhysicalAction rdf:type owl:Class ;
                rdfs:subClassOf :HtnAction .

:PlaceAction    rdf:type owl:Class ;
            rdfs:subClassOf :PhysicalAction ;
            htn_actions:hasEffect :IsOnTopOfEffect ;
            rdfs:label ''{Agent} @Place {Pickable}'' .
```

Listing 6.1: Description of ontology classes in the OWL language using the Turle syntax.

One of the advantages of using action model stored in Ontologenius is the class inheritance. It allows to define properties for one class that will be transmitted to its child classes (*e.g.*, if it exists multiple class representing a Place action, let's say `human_place_cube` and `robot_place_cube`, both inherit from the properties of `PlaceAction` such as the label used for the action verbalization). Another advantage is to be able to link classes through properties and to easily query the KB about it (*e.g.*, what are the effects of the `PickAction` and then what types of effects are they?).

As we know, when an agent performs an action, the other agent may monitor it, if present, in order to follow the task progress and to know when the action is over. A way to know that an action is over is to check if the action effects has been added to the current worldstate or not. However, effects may be perceived differently according to the agent type as humans and robots does not have the same

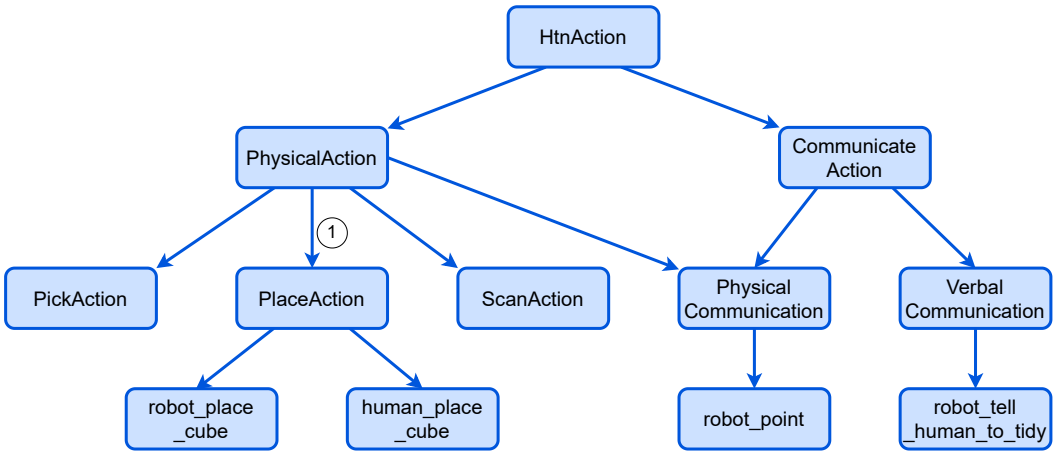


Figure 6.2: The representation of an extract of the ontology class hierarchy graph of HTN actions. Taking the class PhysicalAction, the arrow ① has to be read as *"A PlaceAction is a kind of PhysicalAction"*.

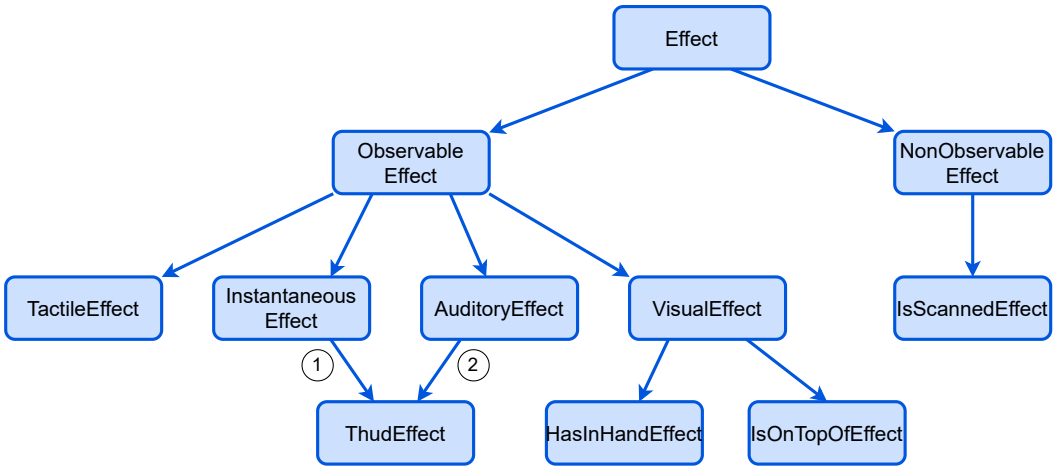


Figure 6.3: The representation of an extract of the ontology class hierarchy graph of action effects. Taking the class ThudEffect, the arrows ① and ② indicate that ThudEffect is and an InstantaneousEffect and an AuditoryEffect.

perception modalities (and even in one given type it can differ). In Figure 6.3, we represented a possible way to model and classify action effects. And so, because Ontologenius is designed for HRI, it is possible to have different representations for robot agents and human agents. We present now a use case with its illustration in Figure 6.4. An agent may have to perform `HeatWaterInKettleAction`. If it is performed by the human, the robot has to monitor the action effects to know if the action is over or not. However, a robot is not able to observe that a kettle has finished to boil water, thus the action has a non-observable effect for the robot. Then, probably the robot will ask the human if the action is over or will see that the human performs its next action of the plan. Now, if we place ourselves in the case

where the robot is the one performing the action – with a smart kettle –, it wants to check if the human could be aware of the action end (because if they are not aware, it should to inform them). The criteria JAHRVIS takes into account is, was the effect observable by the human partner? To answer this question, it first needs to know what the observable effects of `HeatWaterInKettleAction` for the human (if there are). Then, it can query the human's belief base in Ontologenius and get the knowledge that for them, the effect of `HeatWaterInKettleAction` belongs to the class `ThudEffect` (when the kettle stops, it produces a thud) and `TactileEffect` (when the kettle boils water, it becomes hot).

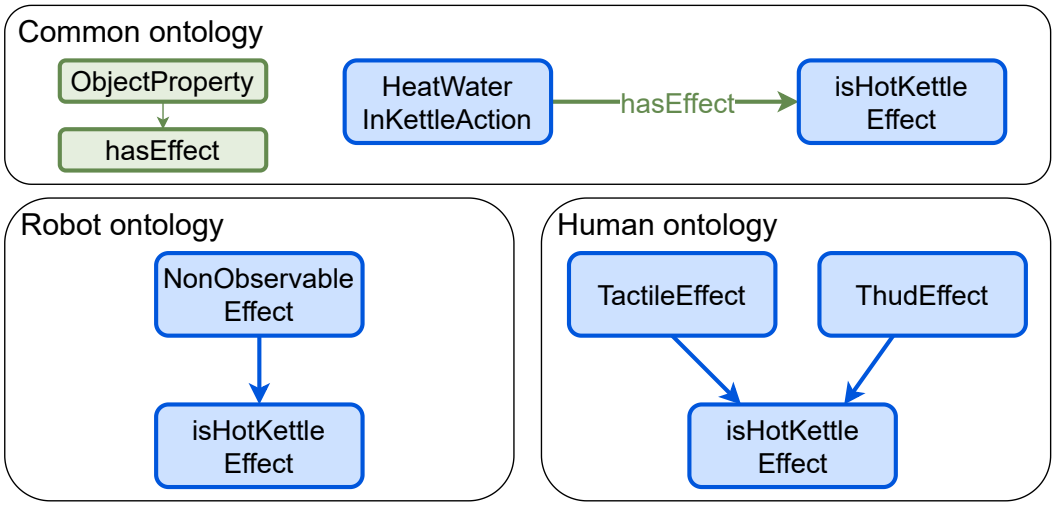


Figure 6.4: Illustration of the human and the robot having differing ontologies. Both have the common knowledge that HeatWaterInKettleAction has isHotKettleEffect as effect. However, in the robot ontology, isHotKettleEffect is defined as a NonObservableEffect whereas in the estimated human ontology it is defined as a TactileEffect and a ThudEffect which are both ObservableEffect as shown in Figure 6.3.

Finally, as mentioned earlier and as visible in Listing 6.1, Ontologenius allows to define label classes and so label for actions. These labels are then used by JAHRVIS to verbalize the plan actions and based on a simple grammar we defined. For example, in "{Agent} @Place {Pickable}", elements between braces are to be instantiated at execution time by JAHRVIS communication process when needed. Then, the @ symbol indicates that the word is a verb that should be conjugated. Verb conjugations can also be found in the KB as shown in Listing 6.2. Thus, the communication manager could process it leading to "I placed the blue cube".

```
:PlaceTSPSimplePresent  rdf:type owl:Class ;
                        rdfs:subClassOf :PlaceSimplePresent ;
                        rdfs:subClassOf :ThirdSingularPersonalForm ;
                        rdfs:label ''place''@fr ;
                        rdfs:label ''places''@en .
```

Listing 6.2: Description of the class describing the verb Place in the third-person present-tense, in the OWL language using the Turle syntax.

We have seen the possibilities offered by Ontologenius to JAHRVIS. Now, we present the two internal action representations, one of the human actions in order to recognize them and the other one is of the robot actions, to allow the robot to estimate the human monitoring activity of its actions.

Internal Action Representation

What does JAHRVIS need to **recognize a human action**? We defined a human action as:

$$Act_H = \langle Action, PrecondL, MoveL, ProgEffectL, NecessEffectL \rangle$$

where $Action$ is a predicate in the form of a triplet $ActName(Agent, Params)$ with $ActName$ the action name, $Agent$ the class of the agent performing it (*e.g.*, Human or Worker) and $Params$ a list of the action parameter classes; $PrecondL$ the list of the action preconditions; $MoveL$ the list of distinctive movements that the human could do when performing the action; $ProgEffectL$ the list of effects that we coined *progression effects* which are action effects, not enough to rule the action end but allowing the plan managers to estimate that an action is progressing towards its end; and $NecessEffectL$ the list of effects that we coined *necessary effects* which are action effects existing iff the action is over.

Our action model takes the form of Jason beliefs written in an ASL file, added as input of the RJA Human Action Monitoring. For example, the actions Pick and Place for a human are represented as:

```
actionModel(pick(Human,[Pickable]),
      [isOnTopOf(Pickable,Support)],
      [handMovingToward(Human,PickableList)],
      [isHolding(Human,Pickable)],
      [~isOnTopOf(Pickable,Support)]).

actionModel(place(Human,[Pickable,Support]),
      [isHolding(Human,Pickable)],
      [handMovingToward(Human,SupportList)],
      [~isHolding(Human,Pickable)],
      [isOnTopOf(Pickable,Support)]).
```

The choice to have two kinds of effects has been made in order to allow the Human Action Monitoring to be robust to a potentially unreliable perception. Indeed, for example in the case of a Place action, the perception of an object hold by a human can be jumpy, and receiving the fact that the object is not in the human's hand anymore. It could reappear a few instant later. On the other, if the robot

perceives that the object has been placed on top of a support, it can assume that the action is really over. The algorithm of Human Action Monitoring will be more detailed in Section 6.3.

The action representation for **robot action execution** allows to match, for a given action, its name and the motion planner and executor needed to execute it. It is possible to specify how the action parameter should be fed to the motion planner and the reaction the robot should have in case of execution failure. The example given in Listing 6.3 shows what functions of the motion planner and executor call and how the system should react in case of failure.

```
// execution limited to 2 trials in a raw
@place[max_attempts(2)]
+!place(Params) : planPick("armUsed", Arm) <-
      .nth(1, Params, Obj);
      headManager(Obj,environment_monitoring,urgent);
      planPlace(Obj, Arm);
      execute("place").

// in case of failure, we try again if did not already
-!place(Params)[error_msg(Msg)] :
  not .substring(max_attempts,Msg) <-
      +error_msg(Msg);
      !place(Params).

// if we tried to execute the plan three times in a raw,
// the action is dropped
-!place(Params)[error_msg(Msg)] :
  .substring(max_attempts,Msg) <-
      ?error_msg(Msg);
      .fail_goal(executeAction,[error_msg(Msg)]).
```

Listing 6.3: Example of an internal action definition for a robot action. The first Jason plan specifies what function to call to execute the Place action. The second plan describes what should be done in case of action planning or execution failure. The third plan is triggered when the first plan has already been tried twice and was requested for a third time. In this case, the failure is signaled at the plan level.

6.1.2 Shared Plan Representation

As explained in Section 5.2, we represent shared plans using Hierarchical Task Network (HTN) as HATP and HATP/EHDA, the planners we used generate HTN-based plans. This formalism allows to deal with goal-based and situation-based activities at different levels of hierarchy such as task, subtasks – abstract tasks using planning vocabulary – and actions – atomic, primitive tasks – and consequently to

consider different levels of granularity. For example, it may be useful to JAHRVIS to be able to request a plan for a given abstract task which failed[1]. Another advantage is that it is easy then for the robot to communicate about subtasks and not only about actions without context. However, according to the task or the domain, the HTN expressiveness for this matter raises discussion. Indeed, HTN plans often make use of recursive abstract tasks which becomes useless for replanning because such abstract tasks have no semantic meaning. Indeed, if we take the piece of plan presented in Figure 6.5, to communicate to the human that the abstract task PlaceAllObjects has failed does not give a information precise enough since there are several of them.

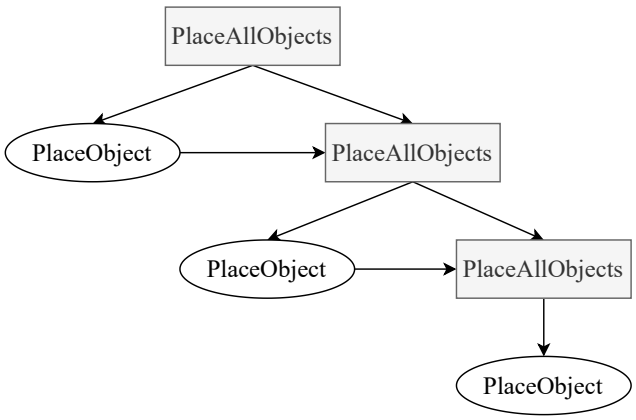


Figure 6.5: Example of a recursive abstract task. The white ellipses correspond to primitive tasks while gray rectangles represent abstract ones.

Moreover, in the next section (see 6.1.3), we show how JAHRVIS feeds the episodic ontology, a timeline, with abstract and primitive task executions. It becomes humanly unreadable with recursive tasks. A concept such as the "iterative task" proposed by Martinie *et al.* [191] would be interesting if used in a HTN plan for HRI. However, there is currently no such thing, so when manipulating such plans we have two modalities:

1. the domain is written being aware of this issue and thus, JAHRVIS takes abstract tasks as input besides primitive tasks (*i.e.*, in the current work, plans generated by HATP/EHDA[2])

2. the domain is written with recursive abstract tasks and thus, JAHRVIS only selects primitive tasks as tasks being part of the plan (*i.e.*, in the current work, plans generated by HATP[3])

We define a shared plan as a sequence of primitive tasks having to be performed

[1]This is future work.
[2]Based on domains intentionally written without recursive tasks by Guilhem Buisan.
[3]Reuse of domains written by Sandra Devin.

by an agent and, abstract tasks. An abstract task λ is defined as:

$$\lambda = \langle id_\lambda, state_\lambda, name_\lambda, \Delta_\lambda \rangle$$

where id_λ is an identification number (id) proper to λ, $state_\lambda$ is the task state estimated by the robot, $name_\lambda$ is the name of the task and the decomposition id $\Delta_\lambda = id_{\lambda\prime}$ with $id_{\lambda\prime}$ the id of the abstract task $\lambda\prime$ that has been decomposed into other tasks, including λ.

And, a primitive task Π is defined as:

$$\Pi = \langle id_\Pi, state_\Pi, name_\Pi, agent_\Pi, params_\Pi, preds_\Pi, \Delta_\Pi \rangle$$

where id_Π is an id proper to Π, $state_\Pi$ is the task state estimation by the robot, $name_\Pi$ is the name of the task, $agent_\Pi$ is the name of the agent that should perform the task, , $params_\Pi$ is the list of parameters required for the task execution, $preds_\Pi = id_{\Pi\prime}, ..., id_{\Pi\prime\prime}$ the list of ids of the tasks $\Pi\prime, ..., \Pi\prime\prime$ needing to be achieved before the task Π can start, and the decomposition id $\Delta_\Pi = id_\lambda$ with id_λ the id of the abstract task λ that has been decomposed into other tasks, including Π.

We defined nine possible values for an abstract or primitive task *state* which are shown in Table 6.2.

State	Description
PLANNED	needs to be done later
TODO	needs to be done now
ONGOING	is in progress
EXECUTED	is achieved
SUSPENDED	needs to be set to UNPLANNED
UNPLANNED	is not part of the plan anymore (used with conditional plans)
NOT_STARTING	was TODO but took to much time before starting
NOT_FINISHED	was started but has not been achieved
NOT_SEEN	was achieved but has not been observed by the other agent

Table 6.2: The nine possible state values of an abstract or primitive task.

So, for example, an excerpt of the BuildingTask plan in which the human and the robot place the first blue cube of the stack and the green cube, generated by HATP/EHDA and represented in Figure 6.6, is:

$\lambda_{13} = \langle 13, \text{PLANNED}, \text{h\_place\_blue\_cube}, 1 \rangle$

$\lambda_4 = \langle 4, \text{PLANNED}, \text{r\_place\_green\_cube}, 1 \rangle$

$\Pi_{139} = \langle 139, \text{PLANNED}, \text{human\_place\_cube}, \text{human\_0}, [\text{blue\_cube\_2,stick}], 138, 13 \rangle$

$\Pi_{141} = \langle 141, \text{PLANNED}, \text{robot\_place\_cube}, \text{pr2\_robot}, [\text{green\_cube,blue\_cube\_2}],$

$139, 4 \rangle$

arrange

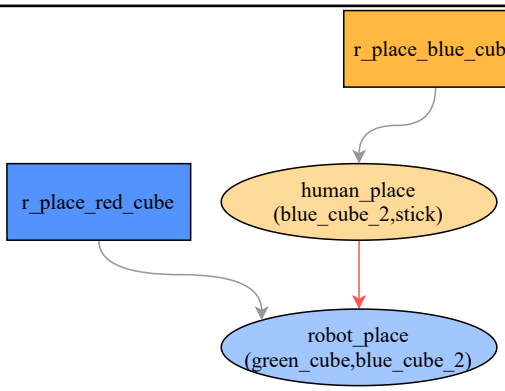


Figure 6.6: An excerpt of the BuildingTask plan. The ellipses correspond to primitive tasks while rectangles represent abstract ones. The blue shapes are robot tasks and the yellow ones are the human ones. Finally, red arrows indicate the sequence of actions in the plans and gray ones represent hierarchical links, *i.e.* the links between the tasks as defined in the HTNs.

6.1.3 Feeding the Knowledge Base

Until now, we stayed focused on semantic knowledge going from the external KB to JAHRVIS. But, as mentioned earlier, one of the external KB is dedicated to episodic knowledge. This KB takes the form of a timeline, managed by Mementar. Whereas Ontologius feeds JAHRVIS with knowledge, the flow is inverted for the episodic data, as Mementar is fed by JAHRVIS among other components. Indeed, when an abstract or primitive task is started or achieved, this information is sent to Mementar for storage with the associated ID and time stamp. The objective is to have a history of the task proceeding. One of the possible use of such a history is for the robot to refer to past events during a task when communicating. We illustrate the timeline in Figure 6.7 with the same example as the one above for the plan, one cube is placed by each agent.

refaire et ajouter le temps en legende

Moreover, JAHRVIS adds the semantic data associated to a task – agent and parameters – to the semantic ontology.

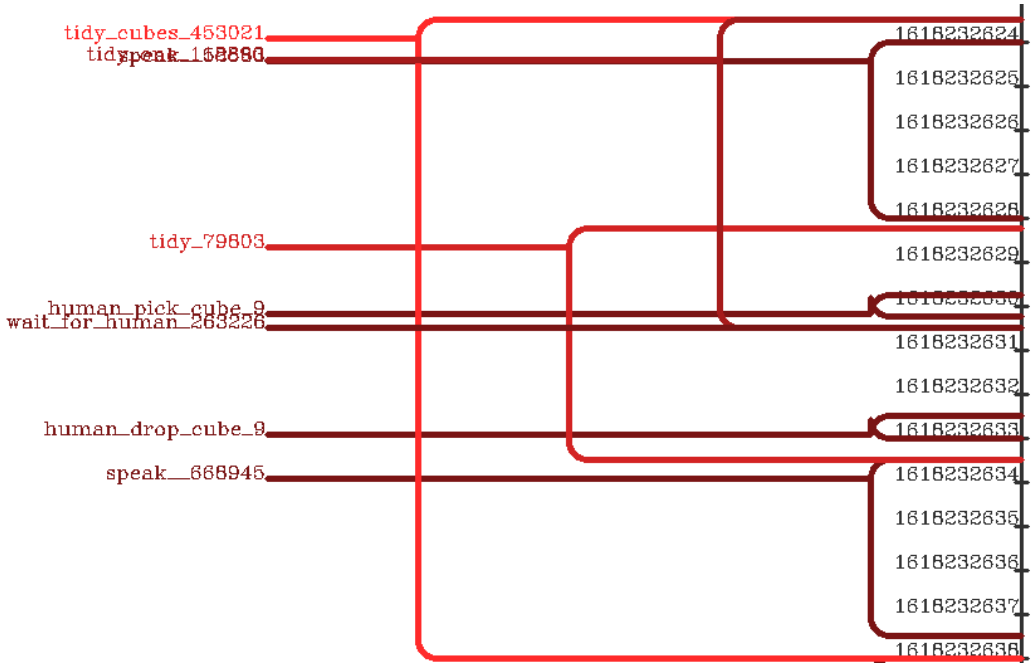


Figure 6.7: Piece of timeline of an executed BuildingTask plan where the robot and the human had to place a cube each. It has been generated by Mementar after having received data from JAHRVIS.

6.2 Interaction Session Management

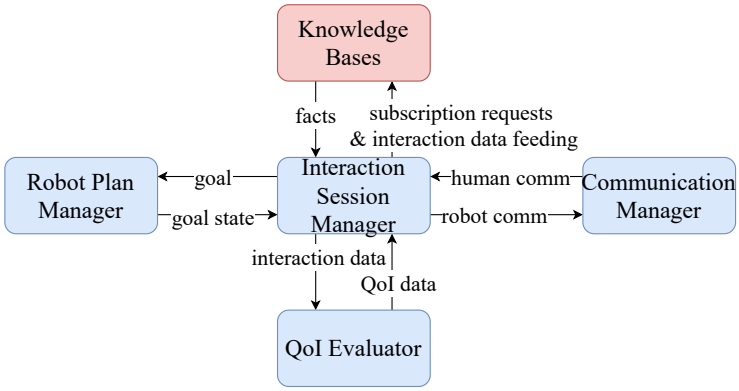


Figure 6.8: The Interaction Session Manager and the RJAs (in blue) and the component of the robotic architecture (in red) with which it interacts.

The Interaction Session Manager (ISM) handles the interaction sessions and manages the robot (shared) goals. Figure 6.8 proposes a focus on the JAHRVIS structure and the components of the robotic architecture around the ISM. Figure 6.9 shows the process we designed, modeling the different states in which the robot can. Moreover, we thought the manager with the ability to consider that a human can

join another one during a conversation. However, we have not implemented it yet.

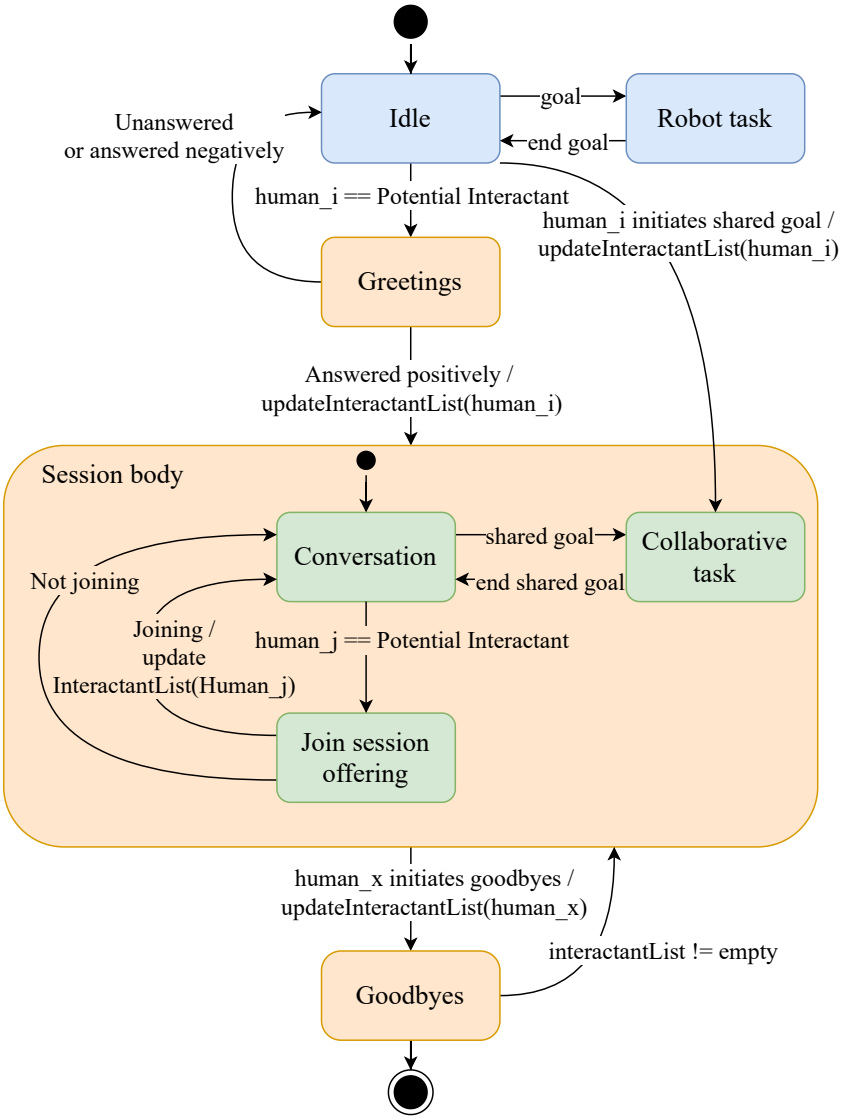


Figure 6.9: States of the Interaction Session Manager (ISM). In blue are represented the states of the ISM when the robot is not in an interaction session. In orange are the states in which the ISM can be during an interaction session. In green are the sub-states of the interaction session body.

A simpler version works as following. An interaction session is triggered when a human is close enough to start a conversation and seems willing to, *i.e.*, when the fact *isEngagedWith*(human_i, robot) is added to the knowledge base and sent to JAHRVIS. In this way, the robot tries to respect people that does not want to interact with it. From there, the robot is in the first phase of the interaction session, the *greetings*. The robot says hello to the human and announces the activities it can perform with them, depending on the context they are in. The interaction

manager triggers the tracking of the human's head by the robot head. This has two purposes: to signal the robot's engagement and to monitor the human's actions. This behavior is quite similar to the one described by Satake *et al.* [247].

An interaction session stays open as long as the human and the robot perform activities together, *i.e.*, as long as the human is engaged in the interaction. This engagement is monitored by the robot in different ways: through the predicate *isEngagedWith*(human_i, robot) during dialogue phases outside a task and through what is happening during a task. If at some point the human is not perceived for a while or the human says goodbye, then the manager ends the session. In the latter case, the robot replies with goodbyes. Finally, it returns to its home-base if it has one.

6.3 Human Actions Recognition

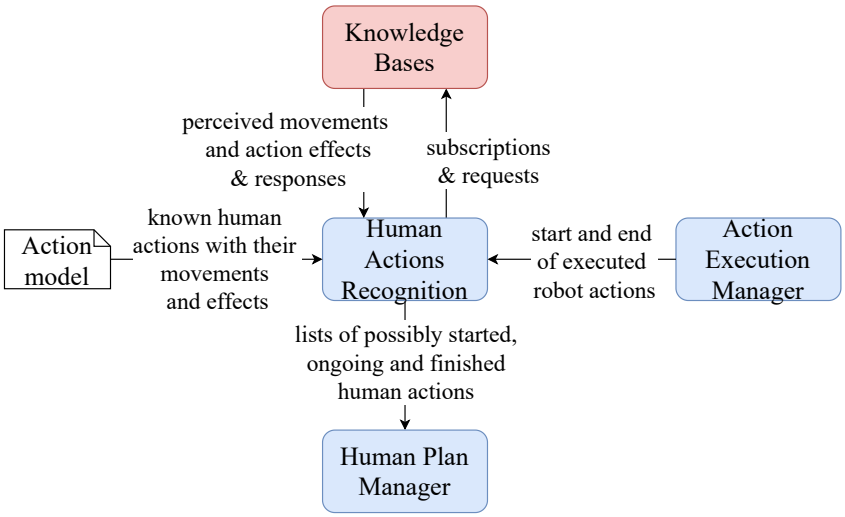


Figure 6.10: The Human Actions Recognition and the RJAs (in blue) and the component of the robotic architecture (in red) with which it interacts. It is fed by a file (in white) with description of the human actions the robot should recognize.

In order to coordinate properly, humans monitor each others when they are in a joint action (see Section 1.3.3.7). The robot needs the same kind of process to be able to assess if the human is doing the action of the plan it expects or not. This allows to follow the plan progress and to estimate the level of human engagement. Existing solutions exist to recognize human actions but none of them matched all our criteria which are:

faire un état de l'art rapide

- it should be easy and quick to add a new action that the robot can recognize
- the process should output the action parameters
- the process should give information about the action progress, *i.e.*, modeling the action start and progression when possible and not only the end
- it needs to be robust to a potentially unreliable perception

- an available open-source code

Thus, we implemented our own model-based solution with an RJA dedicated to Human Actions Recognition (HAR). Figure 6.10 shows its relations with the other RJA of JAHRVIS and components of the robotic architectures. It could be replaced later with a more complex solution meeting the needs, but even though the current one is quite simple it has interested properties, matching the criteria presented above.

The HAR relies on the action model presented in Section 6.1.1 which it loads at initiation. We chose to base our action recognition process on human movements and action effects that the robot can observe. As it needs to recognize them, it extracts the predicate types corresponding to those and subscribes to updates about these facts to Ontologenius as explained in Section 6.1.

Continuously, the HAR RJA receives facts and human movements that are present in the action model, and sends to the Human Plan Manager (HPM) RJA three types data about human actions:

- list of actions that may have started that we coined *possibly started actions*
- list of actions that may be progressing that we coined *possibly progressing actions*
- list of actions that are estimated as finished that we coined *possibly achieved actions*

Action states are updated according to the facts defined in the human action model that the HAR receives. When the state of an action changes to *possibly started*, *possibly progressing* or *possibly achieved*, the affected list is updated and sent to the HPM.

We chose to use the term *possibly* to describe these states as it is well-known that some estimations of action states are false but they allow the system to have an idea of what might be going on.

The algorithm we developed can be depicted in the form of a Finite State Machine representing the state of one action as shown in Figure 6.11 and is implemented in ASL (the code is available in Appendix A.1). Many State Machines can run simultaneously, one for each action that is estimated to be in one of the states. The parameters field of an action are filled as it progresses through the states, according to the movement and effects allocated to this action. timeline

Each transition is triggered by the addition or the deletion of a fact. Using Jason rules[4], facts are analyzed to see if they match a movement or an effect of a known action. For example, if we look at the Place action example presented in Section 6.1.1, when performed by a human, it expects the fact *handMovingToward*(Human, Support) as a movement. Therefore, receiving *rightHandMovingToward*(human_0, placement_1) will match a Place action movement and will add the action to the list of *possibly started actions*. However, receiving *rightHandMovingToward*(human_0, phone) will not, as the phone does not

[4]They are quite similar to Prolog rules.

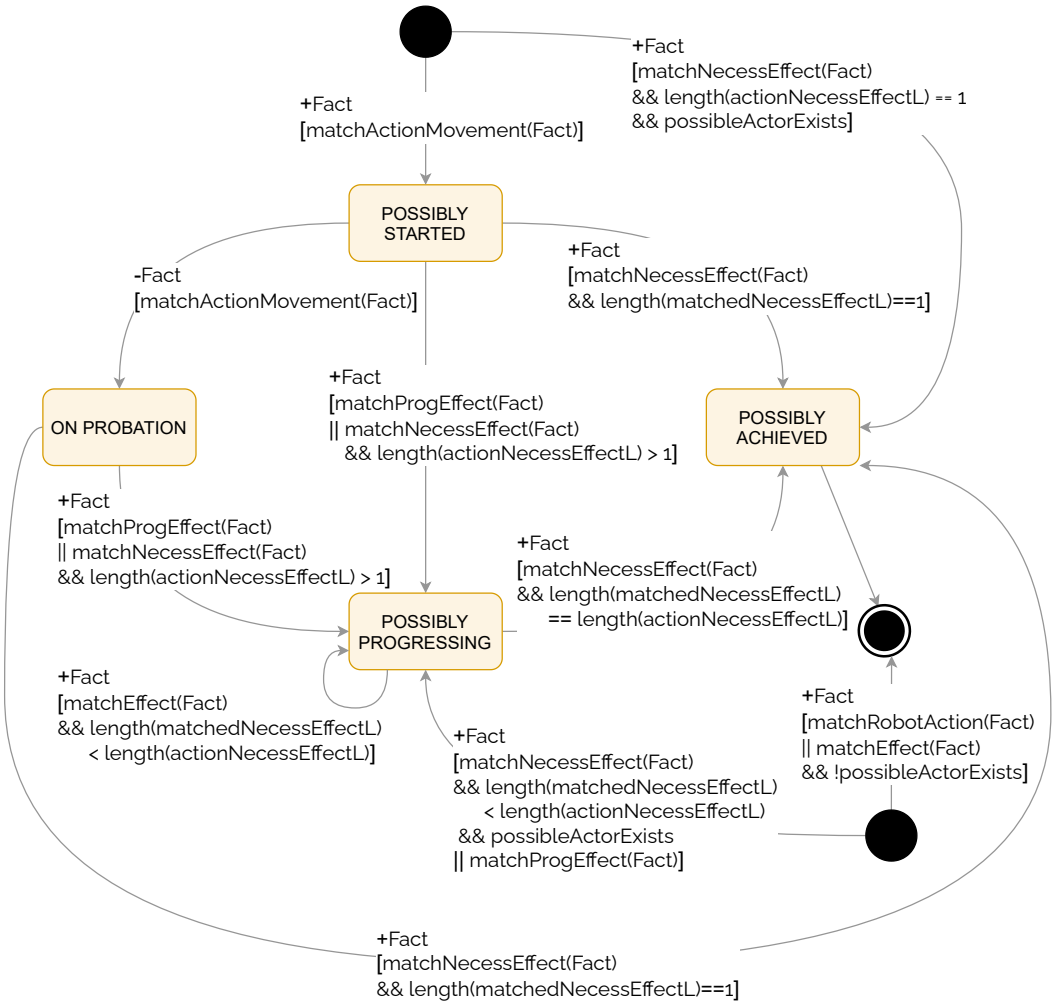


Figure 6.11: Representation of the Human Actions Recognition (HAR) RJA in the form of a Finite State Machine representing the state of one action. Transitions are composed of a triggering event (+Fact or -Fact) and a condition between square brackets. Variable names ending with a "L" are list of effects, either the necessary effect list of a given action in the model (actionNecessEffectL) or the list of facts in the current worldstate corresponding to the effects of a given action (matchedNecessEffectL).

belong to the Support class (but this fact may be useful for recognizing another action).

It is not shown on the Figure for clarity reasons, but each state, except *possibly achieved*, has a transition to the final state which is triggered by a time deadline. Currently, this timeout is the same for each action but as every action type might be of different lasting, a deadline could be specifically set for each one. All the other transitions of the state machine are described in Table 6.3.

Table 6.3: Description of the Finite State Machine shown in Figure 6.11. Inputs are triggering events (+Fact or -Fact) and conditions are between square brackets. Variable names ending with a "L" are list of effects, either the necessary effect list of a given action in the model (actionNecessEffectL) or the list of facts in the current worldstate corresponding to the effects of a given action (matchedNecessEffectL).

Current State	Input and Condition	Next State	Explanation
Initial State	+Fact [matchActionMovement(Fact)]	Possibly Started	When a fact matching a movement and filling the preconditions for a given action is received, HAR computes that the human may have initiated an action.
	+Fact [matchNecessEffect(Fact) && length(actionNecessEffectL) == 1 && possibleActorExists]	Possibly Achieved	The robot may have missed the movement or the progression effect of an action because it was not looking or the perception did not detect it. Then when a necessary effect is received and that only one exists for the action, the action is estimated as achieved if it is possible to find an agent who may have performed it. For now, the finding function is looking for humans in the vicinity of the effect objects, and if there are several humans, it selects the closest one.
	+Fact [matchNecessEffect(Fact) && length(matchedNecessEffectL) < length(actionNecessEffectL) && possibleActorExists matchProgEffect(Fact)]	Possibly Progressing	Similarly to the case above, the robot may have missed the movement or the progression effect of an action. However, if this action has several necessary effects, the action is considered as progressing.
	+Fact [matchRobotAction(Fact) matchEffect(Fact) && !possibleActorExists]	Final State	The HAR is aware of the actions executing by the robot so it does not mismatched its action effects with the ones of another agent. If an effect matches, nothing happen. Likewise if an effect is detected but no agent could be found that might have done it.

Table 6.3: (continued)

Current State	Input and Condition	Next State	Explanation
Possibly Started	+Fact [matchProgEffect(Fact) matchNecessEffect(Fact) && length(actionNecessEffectL) > 1]	Possibly Progressing	When a fact corresponding to a progression effect of the started action is received, or it matches a necessary effect but there are more than one for this action, HAR reinforces its estimation that this action is ongoing by setting it to the progressing state.
	+Fact [matchNecessEffect(Fact) && length(matchedNecessEffectL) == 1]	Possibly Achieved	When a fact corresponding to a necessary effect is received and that there is only one for the started action, the action is considered as achieved as the robot is able to observe its effect and that it had observed the human starting it.
	-Fact [matchActionMovement(Fact)]	On Probation	When a movement fact is removed from the belief base without having observed an effect, it might mean that it was a human hesitation or a false estimation and that the action is not starting. However, it might also be the robot perception being sporadic and so the action goes in this temporary state waiting for a potential coming effect.

Table 6.3: (continued)

Current State	Input and Condition	Next State	Explanation
On Probation	+Fact [matchProgEffect(Fact) matchNecessEffect(Fact) && length(actionNecessEffectL) > 1]	Possibly Progressing	An effect is detected and the action state is resumed.
	+Fact [matchNecessEffect(Fact) && length(matchedNecessEffectL) == 1]	Possibly Achieved	A necessary effect is detected and as there is only one for this action, it is considered as achieved.
Possibly Progressing	+Fact [matchNecessEffect(Fact) && length(matchedNecessEffectL) == length(actionNecessEffectL)]	Possibly Achieved	An necessary effect is received and in total, for this action, there was as many necessary effects received as the ones defined for this action.
	+Fact [matchEffect(Fact) && length(matchedNecessEffectL) < length(actionNecessEffectL)]	Possibly Progressing	When an action effect is received and that either it is another progressing or not the last necessary effect expected for the action, the action state remains progressing.
Possibly Achieved	–	Final State	When an action is estimated as achieved, this is its final state.

When the software starts, the HAR extracts from the internal action representation presented in Section 6.1.1, all the types of facts that should be monitored. Then, it queries Ontologenius to send it each updates about these facts. Thus, when the robot designers decide that a new action should be recognized by the robot, the only thing to add is the action model in JAHRVIS belief base.

Moreover, sometimes new facts are actually effects of robot actions. In order to avoid that robot actions are mistaken for human ones, the AEM signals to the Human Actions Recognition when the robot executes a given action of the plan.

Finally, all the functions to check if a new fact update matches an action effect are Jason rules. They rely on the external knowledge base, here Ontologenius, as there is a need to compare the predicate object and subject expected classes of an action effect with the received ones.

Demo

Now, we give an insight of what happens in the system when the human perform pick and place actions, based on the BuildingTask example. We will present several cases in images and one completed with a timeline and a sequence diagram.

The human has to pick the stick

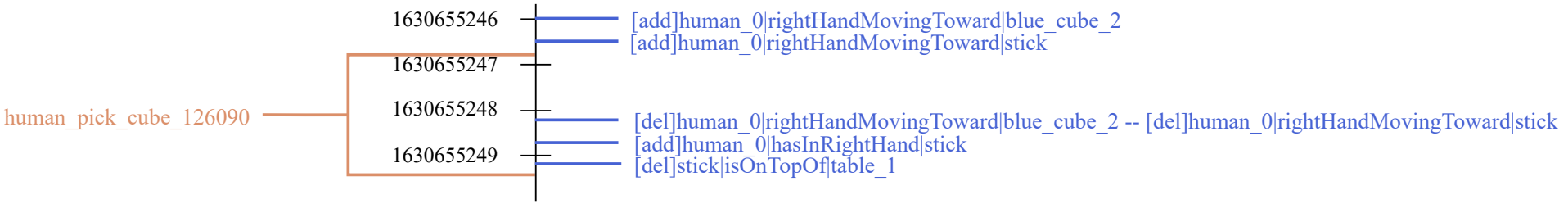


Figure 6.12: Timeline produced by Memetar on which appear facts from perception (blue, on the right) and the action added by the Human Plan Manager based on data from the Human Actions Recognition (orange, on the left). Numbers on the axis are time in milliseconds (epoch time).

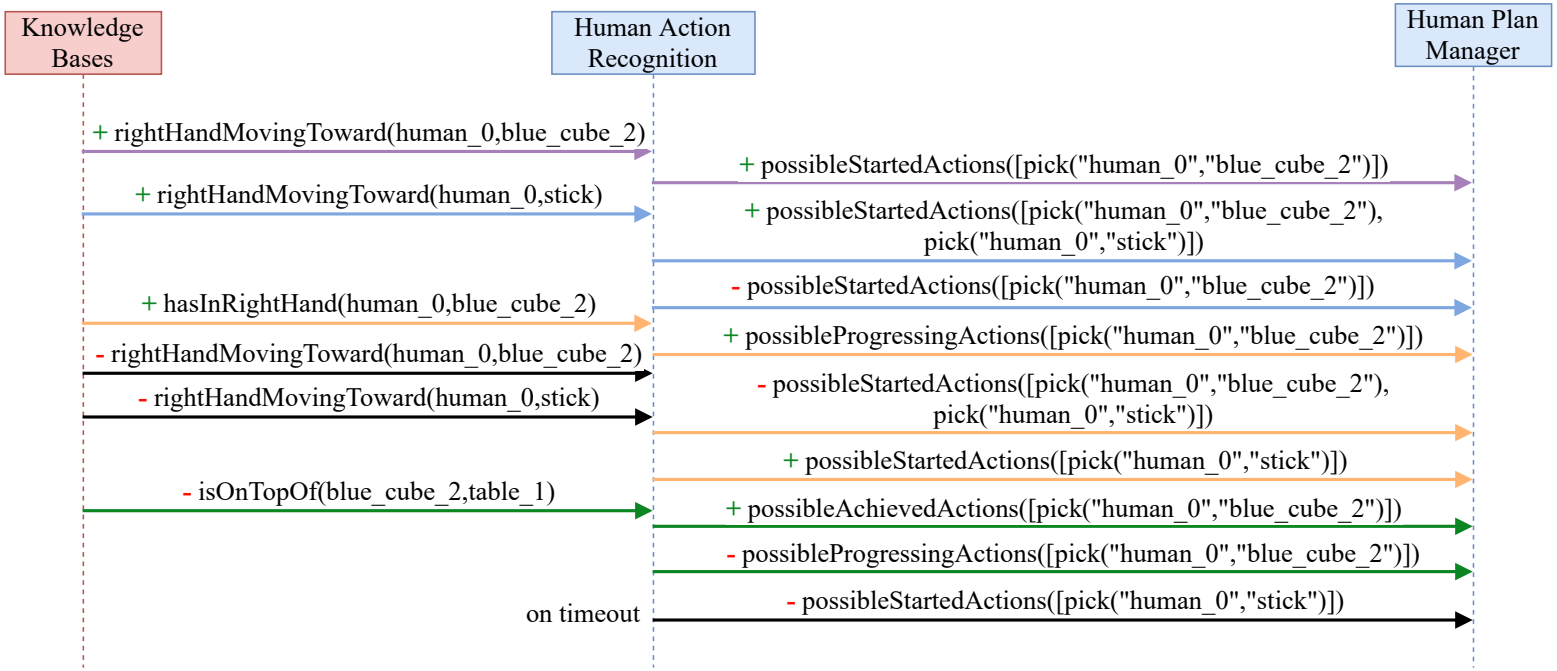


Figure 6.13

6.4 Shared Plans Handling

In order to correctly perform collaborative tasks with humans, the robot needs to know how to perform them. One way is to have a planner with a domain, computing a plan at execution time based on its current knowledge about the environment and interaction. Then, the robot must be endowed with a way to manage the execution of this "recipe". As we place ourselves in the context of joint action, plans manipulated by the robot are shared plans, as presented in Section 6.1.2 (to differentiate from the ASL plans presented in Section 4.3.2 which code all the RJA).

We claim that the robot ability to handle and execute shared plan is enhanced when endowed with Theory of Mind (ToM) (see Sections 1.2 and 2.2.3), as shown by Devin [87]. It allows the robot to be aware of false beliefs or belief divergences in the human's point of view. When such things happen, it can react appropriately, either by acting or communicating. We gave the robot such an ability[5], *via* two processes: the Robot Plan Manager (RPM) and the Human Plan Manager (HPM) (see Figure 5.2). Therefore, the first one handles the robot's beliefs about the plan and the action execution while the second handles the estimation of the human's beliefs about the plan and the communication with the human. The RJAs implementing these processes are presented in Sections 6.4.1 and 6.4.2.

As we designed JAHRVIS to be as generic as possible, it can manage different kinds of human-robot plans as input:

- shared plans in which each action is allocated to an agent as well as action parameters are given objects,
- shared plans in which actions might not be allocated to an agent at planning time and parameters might refer to objects with a semantic query, and
- conditional plans which anticipate different possibilities for the human decision/action.

To generate these plans, we worked with two planners, HATP and HATP/EHDA presented in Section 3.3.3.4.

"Usual" shared plans with HATP and HATP/EHDA The first type of shared plans handled are what we could call "usual" shared plans. Each action is allocated to an agent as well as action parameters are given objects. Thus, in this kind of plan, no decision is left to JAHRVIS about who should execute the action or with what object.

AgentX shared plans with HATP The second type of shared plans is an extension of the work of Devin about postponing some decisions from planning time to execution time about the actor of some actions and some parameters [88]. In works previous to Devin, all the actions of the computed plans were allocated and completely instantiated during plan elaboration.

[5]The robot has a first-order ToM, it estimates the human knowledge about the task but it does not compute what the human thinks of the robot knowledge about the task (second-order ToM)

We re-implemented her idea of *AgentX* in our plan managers (with some modifications, for example we do not replan once an action is allocated as we are able to identify in real-time if a next action is still feasible or not), enabling the *choice* of the agent who should perform the action at execution time when the planner has computed that both agents could do it. This a mean to specify a goal in a more abstract way. Thus, when an action can indifferently be done by both agents, the planner returns $\Pi = \langle id_\Pi, state_\Pi, name_\Pi, \text{AgentX}, params_\Pi, preds_\Pi, \Delta_\Pi \rangle$.In this case, according to what JAHRVIS estimates the human wants to do, it can allocate the action to itself or to them. Therefore, in the BuildingTask example, instead of having the planner arbitrary choosing which agent, the human or the robot, will place the first blue cube on the stick, it allocate it to AgentX.

Then, a similar idea has also been developed by Devin for the parameters. Indeed, with usual HATP plans, still with the BuildingTask example, the planner would have generated a plan where it is already decided that the robot should place, for example, its red cube on the placement 1 and the human should place theirs on the placement 2. We could have:

$$\Pi_1 = \langle id_{\Pi_1}, \textsc{Planned}, \text{human\_place\_cube}, \text{human\_0}, [\text{red\_cube\_2,placement\_2}], preds_{\Pi_1}, \Delta_{\Pi_1} \rangle$$
$$\Pi_2 = \langle id_{\Pi_2}, \textsc{Planned}, \text{robot\_place\_cube}, \text{pr2\_robot}, [\text{red\_cube\_1,placement\_1}], preds_{\Pi_2}, \Delta_{\Pi_2} \rangle$$

However, actually, it does not matter here where which agent place their red cube. Thus, Devin introduced the use of the notion of object similarity: two *similar* objects will have the same role in the task, they are functionally equivalent. With this new notion, instead of having the planner arbitrary deciding which individual should be used when two of them are equivalent, it manipulates object high level names. Thus, in the BuildingTask, for the two first actions, we have:

$$\Pi_1 = \langle id_{\Pi_1}, \textsc{Planned}, \text{human\_place\_cube}, \text{human\_0}, [\text{red\_cube\_2,placement}], preds_{\Pi_1}, \Delta_{\Pi_1} \rangle$$
$$\Pi_2 = \langle id_{\Pi_2}, \textsc{Planned}, \text{robot\_place\_cube}, \text{pr2\_robot}, [\text{red\_cube\_1,placement}], preds_{\Pi_2}, \Delta_{\Pi_2} \rangle$$

To have more expressiveness, we brought a new modification to the plans returned by HATP, in collaboration with Guillaume Sarthou. Indeed, instead of returning an object generic name, it returns a SPARQL query with the constraints used in the domain. Thus, keeping the same example as previously, the robot and

human actions become:

$\Pi_1 = \langle id_{\Pi_1}, \text{PLANNED}, \text{human\_place\_cube}, \text{human\_0},$
$[\text{red\_cube\_2}, \text{?0 isA Placement NOT EXISTS \{ ?0 isUnder ?2. ?2 isA Cube \}}],$
$preds_{\Pi_1}, \Delta_{\Pi_1} \rangle$

$\Pi_2 = \langle id_{\Pi_2}, \text{PLANNED}, \text{robot\_place\_cube}, \text{pr2\_robot},$
$[\text{red\_cube\_1}, \text{?0 isA Placement NOT EXISTS \{ ?0 isUnder ?2. ?2 isA Cube \}}],$
$preds_{\Pi_2}, \Delta_{\Pi_2} \rangle$

This allows the RPM to directly request Ontologenius to get an object list matching this query and to select an object among it at execution time. And, when the human performs an action with a SPARQL query as parameter, the HPM can check if the object on which the human is acting matches the query. We can see that this solution is enhanced compared to the one presented by Devin. Indeed, `red_cube` does not tell that it should be reachable by an agent and that it should not be on the top of another cube yet but `?0 isA Cube. ?0 hasColor red. ?0 isReachableBy ?1 NOT EXISTS { ?0 isOnTopOf ?2. ?2 isA Cube }` does. For example, in Devin, the reachability test was written in the plan manager of the supervisor, in a hard-coded manner.

The plan for the BuildingTask example is presented in Figure 6.14.

Conditional shared plans with HATP/EHDA Finally, the last type of shared plans we manipulated is conditional plan, generated by Human Aware Task Planner with Emulation of Human Decisions and Actions (HATP/EHDA) [53]. It is another mean to postpone decision at execution time about an agent actor or parameters, with plans where branch junctions concern human decision. Moreover, it gives a better insight about the human's choices and decisions as they are formalized with the plan. For example, in Figure 6.15, is shown a conditional plan computed by HATP/EHDA[6]. The plan indicates that the robot has to ask for human help. Then, the human will either get the coffee or fill the water. Thus, at planning time, the choice of human actions is not made, but thanks to the conditional plan, both possible solutions are planned and it is up to JAHRVIS to follow the proper one depending on the human action detected during execution.

put on left and right pages

[6]The domain for this plan was written by Guilhem Buisan

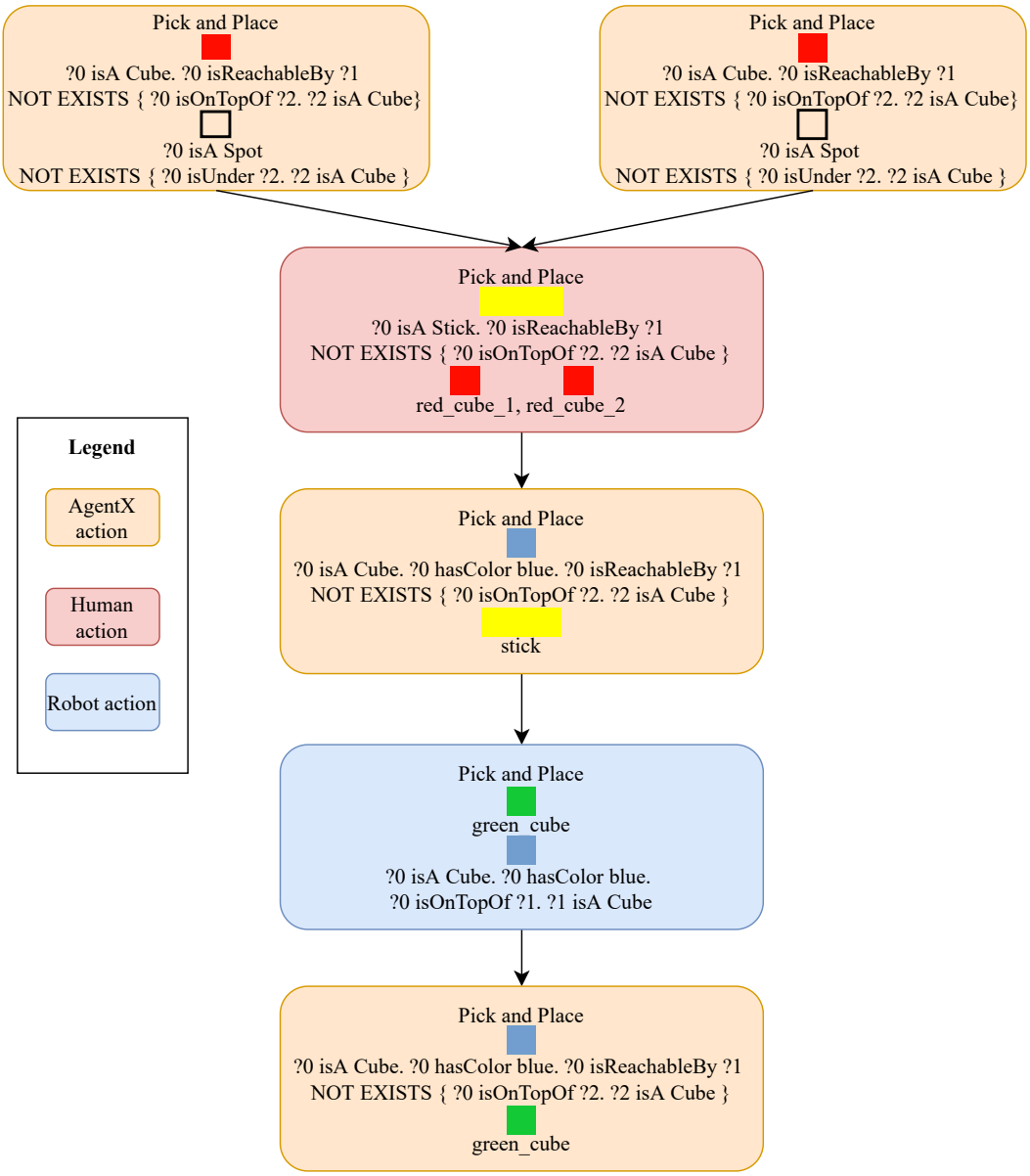
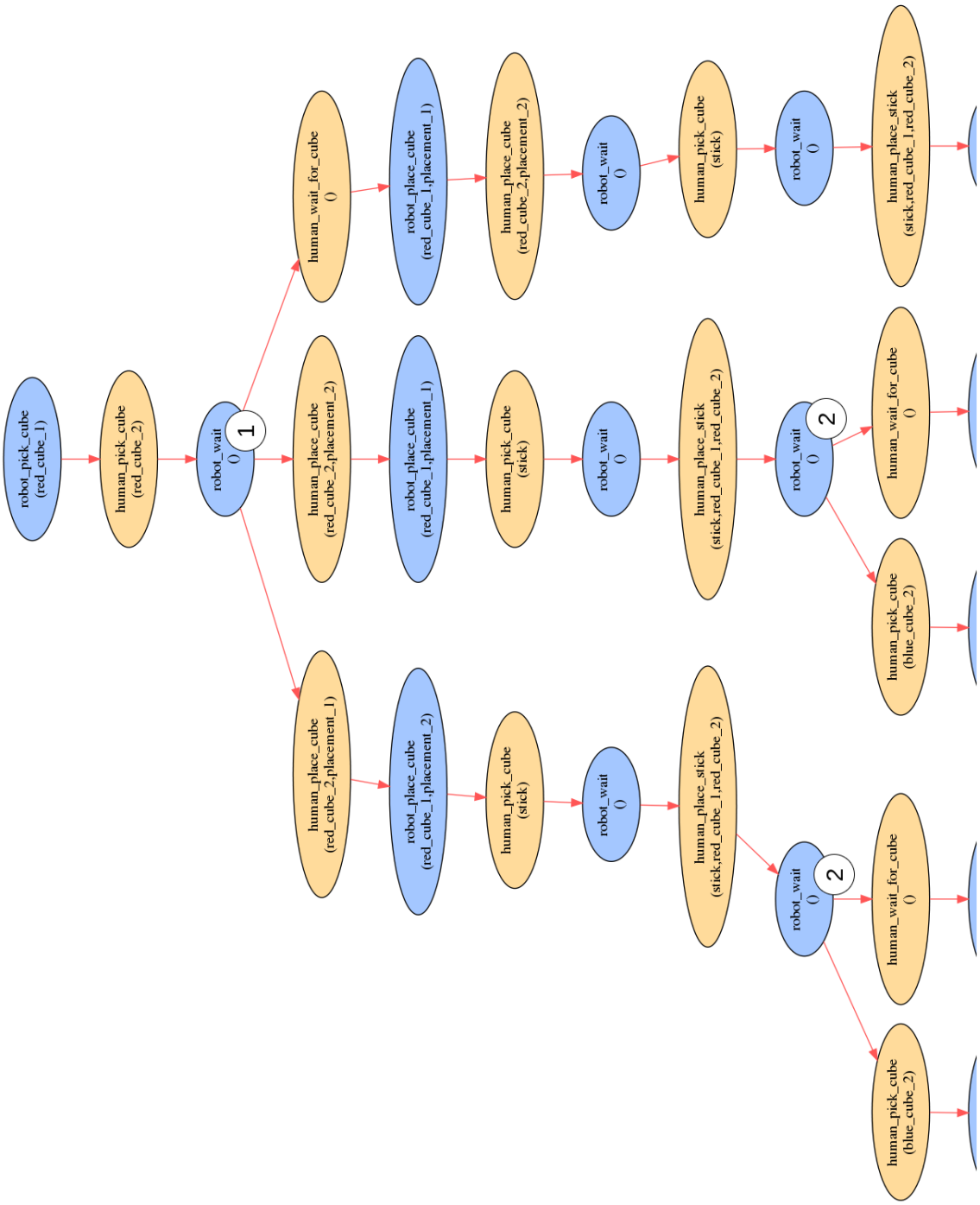


Figure 6.14: A shared plan for the BuildingTask computed by HATP with modifications to generate SPARQL queries instead of object names as action parameters.


robot_pick_cube (red_cube_1)
human_pick_cube (red_cube_2)
robot_wait ()
1
human_wait_for_cube ()
robot_place_cube (red_cube_1,placement_1)
human_place_cube (red_cube_2,placement_2)
robot_wait ()
human_pick_cube (stick)
robot_wait ()
human_place_stick (stick,red_cube_1,red_cube_2)
human_place_cube (red_cube_2,placement_2)
robot_place_cube (red_cube_1,placement_1)
human_pick_cube (stick)
robot_wait ()
human_place_stick (stick,red_cube_1,red_cube_2)
robot_wait ()
2
human_wait_for_cube ()
human_pick_cube (blue_cube_2)
human_place_cube (red_cube_2,placement_1)
robot_place_cube (red_cube_1,placement_2)
human_pick_cube (stick)
robot_wait ()
human_place_stick (stick,red_cube_1,red_cube_2)
robot_wait ()
2
human_wait_for_cube ()
human_pick_cube (blue_cube_2)

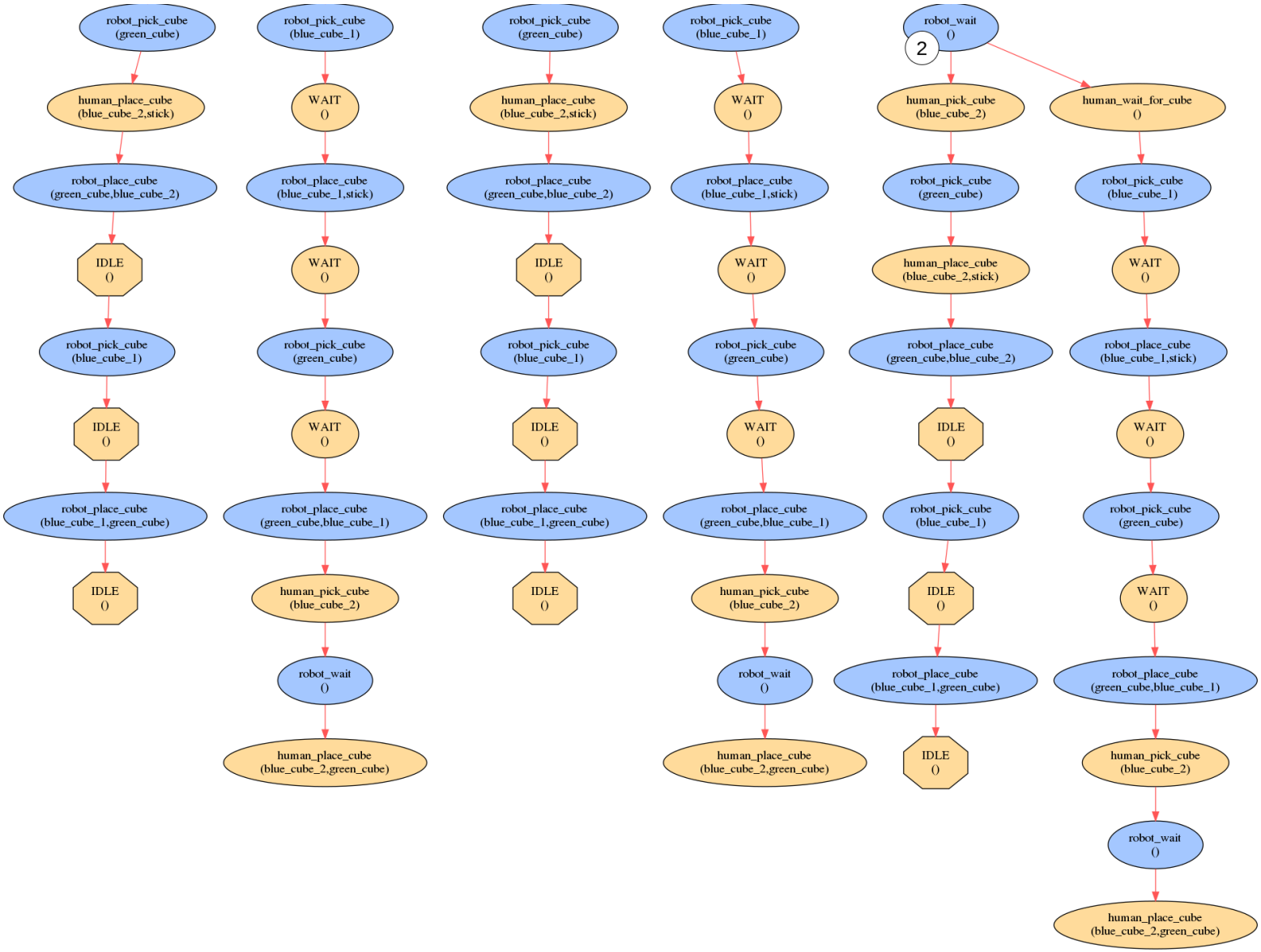


Figure 6.15: A conditional plan for the BuildingTask computed by HATP/EHDA. Twice during the task, the human has a choice. First, they can choose where to place their red cube or to wait for the robot to choose for the placement, ① on the figure. The second choice happens for the positioning of the first blue cube of the stack, either the human can place it on the stick or leave it to the robot, ② on the figure.

JAHRVIS could be used to execute plans from other HTN planners than HATP and HATP/EHDA by adding a Java class to format abstract and primitive tasks as presented in Section 6.1.1 – HATP and HATP/EHDA have a dedicated Java class each, for action formatting, but their plans are handled with the same code in the plan managers.

Now, we will present the two processes in charge of the shared plan management, one to handle the plan on the robot side, *i.e.*, its updates and the action execution, and the other one to handle the estimated human mental states about the shared plan. When either the robot or the human starts and finishes an action execution, facts corresponding to these events, are added to Ontologenius and Mementar to keep track of what happened during the interaction. It also registers the data about the abstract task. Then, a component of the robotic architecture used to generate communication about elements, the Referring Expression Generator (REG), can use such information when querying by JAHRVIS to refer to the past (*e.g.*, "the cube you took").

When we will mention the robot monitoring with its head, it is done through a component described in Section 6.1.1.

replan,contingencies?

6.4.1 Robot Plan Management

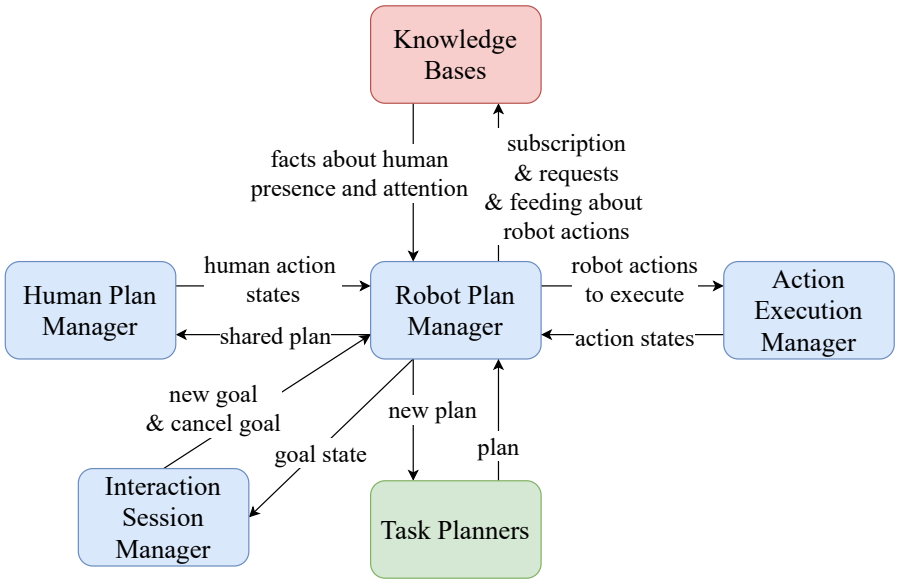


Figure 6.16: The Robot Plan Manager and the RJAs (in blue) and the components of the robotic architecture (in red and green) with which it interacts.

As explained earlier, there are two processes to manage the shared plans. One of them is the Robot Plan Manager (RPM). It is in charge of the plan updates, maintaining the robot knowledge about the ongoing goal, and deciding which action

should be performed by the robot and when. Figure 6.16 shows its relations with the other RJA of JAHRVIS and components of the robotic architecture.

When it receives a new goal from the Interaction Session Manager, it queries the Task Planner for a plan. This plan is a sequence of abstract and primitive tasks as described in Section 6.1.2. First, when receiving the plan, and then at each end of action execution, the abstract and primitive task states of the plan are screened in order to find the next primitive task to perform. The found ones have their state set to TODO. The implemented algorithm to do so is presented in Algorithm 1.

Algorithm 1 Update of a plan

```
function UpdatePlan
    for each Π_i with state_i = PLANNED in Plan do
        preds :=FindAll(id_j)
                                such as Π_j ∈ Plan, id_j ∈ preds_i, state_j ≠ EXECUTED
        if preds = ∅ then
            state_i := TODO
            UpdateAbstractTasksToOngoing(Δ_i)
    if ∀Π_x ∈ Plan, state_x = EXECUTED or state_x = UNPLANNED then
        GoalState :=SUCCEEDED

function UpdateAbstractTasksToOngoing(λ_i)
    if ∃λ_i with state_i = PLANNED then
        state_i := ONGOING
        UpdateAbstractTasksToOngoing(λ_Δi)
```

As we used Jason, a process can react to events (see Section 4.3.2). Every abstract or primitive task state update triggers an event. We represented what happened when a primitive task is updated to TODO in Algorithm 2. There are three possible cases, either an action is to be performed by the human, or the robot, or it is undefined which is represented by the AgentX.

When an action should be performed by the human and if the robot does not have an action to do at the same time, the robot has to monitor them, or rather the parameters of their action, in order to be aware of what they are doing. To monitor, robot can use several modalities when they exist. Indeed, it can monitor with the camera on its head but also with a (lidar) scanner for example. As the most important thing for the RPM is to monitor parameters and not how to monitor them, it sends to a component in charge of the robot resources the objects of interest for the given action. For now, only one parameter among the parameter list is selected to be monitored. The function to select this object of interest among the action parameters is simple, we take the first of the list, but it could be refined. Or, as explained earlier, parameters can be in the form of SPARQL queries. If it is the case, the robot chose to monitor the human closest one among the ones returned by Ontologenius. Then, it waits an update on the action state – which is updated by the Human Plan Manager (HPM) and then sent to the RPM.

Algorithm 2 Event action todo in RPM

```
function On(Π_i) with state_i =TODO
    if agent_i ∈ Human and name_i ∈ PhysicalAction then
        objM := ChooseObjectToMonitor(params_i)
        SetMonitorObject(objM)
        WaitForPrimTaskStateToChange(Π_i)
    else if agent_i ∈ Robot or agent_i ∈ AgentX then
        if ∃p ∈ params_i, p is a SPARQL query then
            oneAgentOnly, newParams := InstantiateParams(Π_i)
            params_i := newParams
        if oneAgentOnly ≠ ∅ then
            agent_i := oneAgentOnly          ▷ Triggers a new On function as
                                               TODO action updated
        else
            AllocatePrimTask(Π_i)
            if agent_i ∉ Human then
                SendMessage(ActionExecutionManager,Π_i)

function InstantiateParams(Π_i)
    for each p in params_i do
        if p is a SPARQL query then
            sparqlQ := SparqlToElementList(p)   ▷ sparqlQ is a list of list.
            oneAgentOnly = ∅
            if ∃a ∈ sparqlQ, a ∈Agent and a is unique then
                oneAgentOnly := a
            add sparqlQ in newParams
        else
            add p in newParams
    return oneAgentOnly, newParams
```

When an action should be performed by the robot or the AgentX, first it needs to check if all action parameters are already instantiated and not a SPARQL request. When a parameter is a SPARQL request, it queries Ontologenius to get all the objects matching it, and eventually, the agents, in the form of a list of list. For example, if the SPARQL request is `?0 isA Cube. ?0 hasColor red`, then the result could be `[[red_cube_1],[red_cube_2]]`. Or, in case where there is another element in addition to the object, such as an agent, *e.g.*, `?0 isA Cube. ?0 hasColor red. ?0 isReachableBy ?1`, the result could be `[[red_cube_1,robot],[red_cube_2,human]]`. Sometimes, an agent action is allocated to the AgentX but the environment may have changed since planning time. Then, there may be one agent only, either the human or the robot, returned in the object list. In this case, the agent action value is updated with this agent. Next, the action has to be allocated to an agent. If the agent value corresponds to the robot, the only thing to do is to select the parameters to execute the action in case some

of them are object list. For now the function is simple, the robot choses the first one of the list. If the agent value corresponds to the AgentX, then the RPM checks if another action exists with the same parameters and the TODO state. Indeed, the human cannot perform two actions at the same time so the RPM can allocated one to the robot. In case there is no other action, as we think the robot as a human helper, it should leave the choice to them if they want to perform the action or not. Devin showed that naive users preferred when the robot asked them what they wanted to do [88] but it has been shown that regular users preferred when a robot adapted to them. Therefore, we chose the adaptive option, where the robot waits a few seconds to see if the human starts to perform the action. If they do, the action is allocated to the human and if they do not, the RPM allocates the action to the robot. Finally, when an action was allocated to the robot, it is sent to the Action Execution Manager that will handle the action execution as indicated by its name.

add ref

Algorithm 2 Event action todo in RPM(continued)

```
function AllocatePrimTask(Π_i)
    if agent_i ∈Robot or (∃Π_j with j ≠ i, state_j =TODO, params_i = params_j,
                                                    agent ∈ AgentX) then
        AllocatePrimTaskToRobot(Π_i)
    else
        while t_current < T_max_wait or
              (state_i = ONGOING or state_i = EXECUTED) and agent ∈ Human do
                                  ▷ Π_i state may be updated by the Human Plan Manager
        if agent ∉ Human then
            AllocatePrimTaskToRobot(Π_i)

function AllocatePrimTaskToRobot(Π_i)
    for each p in params_i do
        SelectParams(params_i)
        agent_i ≔ Robot
```

We represented what happened when a primitive task is updated to EXECUTED in Algorithm 3. As explained above, the manipulated plans can be conditional plans. Thus, at the end of each action execution, the RPM looks for if actions of other branches exist. First, it tries to find if the agent and predecessors of the action which just finished are the same than the one another action. If it is the case, all the abstract and primitive task descendant of this found action are set UNPLANNED.

Algorithm 3 Event action executed in RPM

function On(Π_i with $state_i$ =EXECUTED)
  EndObjectMonitoring($params_i$)
  RemoveParallelBranches($agent_i$,$preds_i$)
  UpdateAbstractTaskState(Δ_i,EXECUTED)
  UpdatePlan ▷ see Algorithm 1

function RemoveParallelBranches($agent_i$,$preds_i$)
  $primTasksToUnplan$:= FindAll(Π_x) with $agent_x = agent_i, preds_x = preds_i$, ($state_x$ = TODO or $state_x$ = SUSPENDED)
  RemovePrimTasks($primTasksToUnplan$)

function RemovePrimTasks($primTasksToUnplan$)
  for each Π_i **in** $primTasksToUnplan$ **do**
    $state_i$:= UNPLANNED
    UpdateAbstractTaskState(Δ_i,UNPLANNED)
    RemoveChild(Π_i)

function RemoveChild(Π_i)
  $primTasksToUnplan$:= FindAll(Π_j) with $id_i \in preds_j$
  RemovePrimTasks($primTasksToUnplan$)

function UpdateAbstractTaskState(id_x,$newState$)
  if $\forall\lambda_i$,Π_i with $\Delta_i = id_x$, ($state_i$ = EXECUTED or $state_i$ = UNPLANNED) **then**
    $state_x$:= $newState$
    UpdateAbstractTaskState(Δ_x,$newState$)

6.4.2 Human Plan Management

add wait and check already performed (state based + transition based)

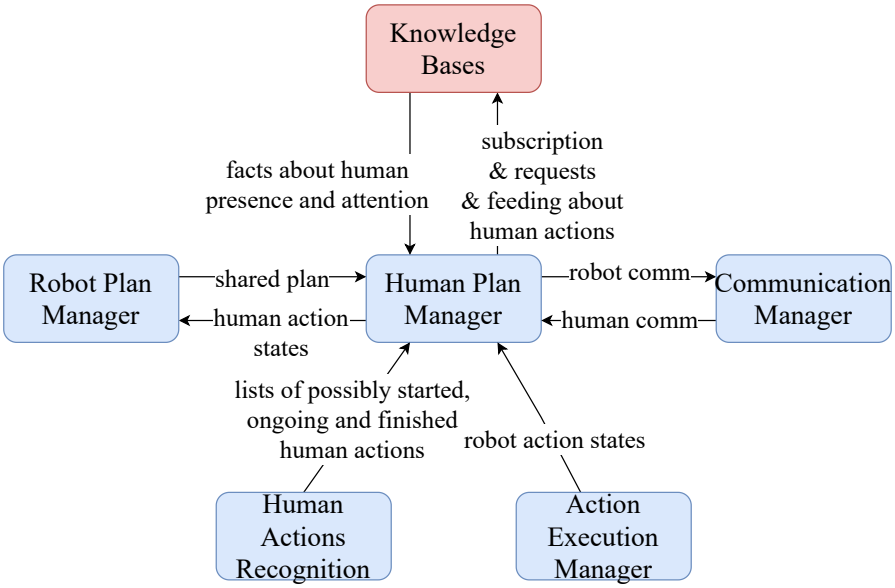


Figure 6.17: The Human Plan Manager and the RJAs (in blue) and the component of the robotic architecture (in red) with which it interacts.

The Human Plan Manager (HPM) keeps track of the estimated human mental state about the ongoing shared plan, endowing the robot with Theory of Mind (see Section 1.2). The role of this process is central, as it receives the data about the recognized human actions, deduces what the human might or might not know about the plan or action executed by the robot, and requests the communication to perform to the Communication Manager (CM). Figure 6.17 shows its relations with the other RJA of JAHRVIS and components of the robotic architecture.

When a shared goal starts, it receives from the RPM the list of primitive tasks composing the plan. The action states are updated with the same algorithm as the RPM, Algorithm 1 (with the function updating the abstract tasks being not used). We distinguish two cases of TODO actions, the actions to be performed by the human and the ones to be performed by the robot.

Action to be performed by the human We present Algorithm 4, describing what happens in the HPM when it computes that the human has an action to perform, with effects that are VisualEffects (see Section 6.1.1). It only shows when everything goes well, *i.e.*, that the human perform the right action in the allocated time. The way the robot reacts if the human is not acting, by communicating, is described with Algorithm 5 for the case where the action never started and with Algorithm 6 for the case where the robot detected the start but cannot see the final action effects.

As we can see in Algorithm 4, the HPM estimates the human is aware that they have an action to perform, it checks if action data received from the Human Actions Recognition matches it. The function comparing the TODO action with monitored actions communicated by the HAR is a Jason rules, comparing the agent names, the action classes and the parameters. If the TODO action has SPARQL-like parameters, the rule allows to check if it can corresponds to the parameters of a given recognized action. Moreover, because JAHRVIS enables the use of conditional plan where the branch choices are made by the human, when the human makes one of this choice, the other branches have to been SUSPENDED and then UNPLANNED.

Algorithm 4 Event action todo by human in HPM

```
function On(Π_i) with agent_i ∈ Human and state_i = TODO
    matchingAction := false
    while t_current < T_timeout,start and not matchingAction do
        if Match(Π_i, monitoredAction_STARTED,PROG,ACHIEV) then
            matchingAction := true
    if t_current > T_timeout,start then return
    if Match(Π_i, monitoredAction_STARTED,PROG) then
        state_i := ONGOING
        SendMessage(RobotPlanManager, Π_i)
        matchingAction := false
    primTasksToSuspend := FindAll(Π_j) with id_i ∈ preds_j
    SuspendPrimTasks(primTasksToSuspend)
    while t_current < T_timeout,achiev and not matchingAction do
        if Match(Π_i, monitoredAction_ACHIEV) then
            matchingAction := true
    if t_current > T_timeout,achiev then return
    state_i := EXECUTED
    SendMessage(RobotPlanManager, Π_i)
    RemoveParallelBranches(human, preds_i)        ▷ see Algorithm 3
    UpdatePlan                                    ▷ see Algorithm 1
```

When the robot estimates that the human knows they should perform an action but this does not happen, it initiates a communication through the Communication Manager in order to indicate to the human that they have this given action to do. Thus, it updates once again its estimation of the human mental state about the action, setting it to TODO since it informed them. It is described by Algorithm 5.

A bit similarly, when the robot observed the beginning of an action, it asks the human if they did it, as it might have missed the end of the action execution and/or for some reason might not be seeing the action necessary effect. If the human answers yes, the robot updates the action state as well as the actions effects in Ontologenius. In the other case, the robot set the action to TODO as the human knows they should do it. This function could be enhanced with a more sophisticated dialog.

pas codé

Algorithm 5 Handling of action todo timed out on wait for started/progressing action by human in HPM

```
function NotDoing(List of Π with state = todo)
    if first time for these actions then
        for each Π_i in List of Π do
            state_i := not_starting
        SendMessage(CommManager,List of primTask)
        for each Π_i in List of Π do
            state_i := todo
    else
        Negociation or StopGoal                ▷ Negociation not implemented
```

Algorithm 6 Handling of action todo timed out on wait for achieved action by human in HPM

```
function NotDoing(Π_i) with agent_i ∈ Human and state_i = todo
    if first time for this actions then
        state_i := not_finished
        answer := SendMessage(CommManager,Π_i)
        if answer = no then
            state_i := todo
        else
            state_i := executed
            UpdateOntologenius(necessEffectL_i)
    else
        Negociation or StopGoal
```

Action to be performed by the robot Now, the HPM should also handle when an action is to be performed by the robot. Thanks to the class action we defined (see Section 6.1.1), actions on the environment and communication actions can be process differently. For the latter, human attention is monitored by the Communication Manager.

The handling of an action performed by the robot depends on the estimated establishment of a (simple) joint attention (see Section 1.3.3.5) between the human and the robot. An activity diagram presented in Figure 6.18 shows that when the HPM is informed by the Action Execution Manager (AEM) of a robot ongoing action, it monitors, if the human is in its field of view, their attention towards the action parameters or the robot. When the HPM estimates that the human sees what is going on, then it updates the human's mental state about this action. When it estimates that they have not seen the action, then it considers that the human has a false belief about the action, as in the robot's belief base the action is executed but not in the human's one, there is a belief divergence (see Section 1.2). Thus, it communicates to realign the human's beliefs. Moreover, even if the robot was in the human's field of view (FoV), sometimes some action effects are non observable

(see Section 6.1.1), so this is another case where the robot will communicate about an action it executed. Then, when an action is set as EXECUTED in the human's mental state, it updates the plan with the function presented in Algorithm 1.

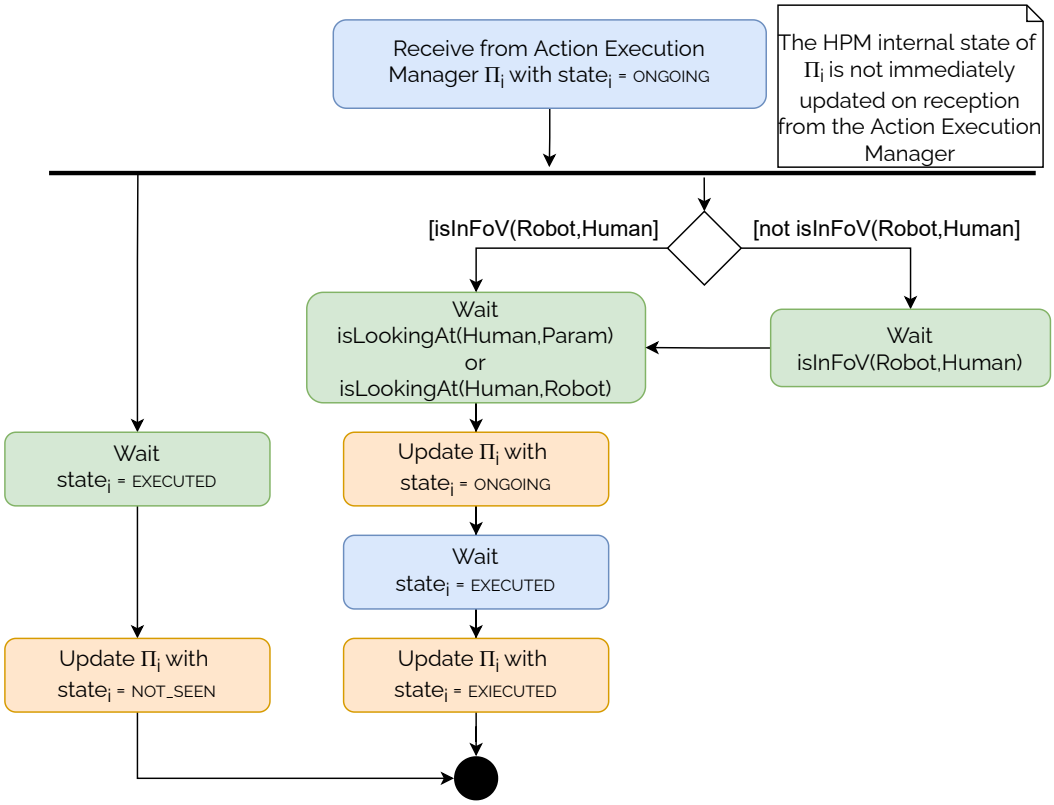


Figure 6.18: Activity diagram representing what happens when the Action Execution Manager (AEM) sends to the HPM that the robot started to execute an action. We represent in blue the nodes receiving data from the AEM, in green the ones receiving data from Ontologenius and in orange the ones updating the action state in the HPM belief base.

6.5 Action Execution Management

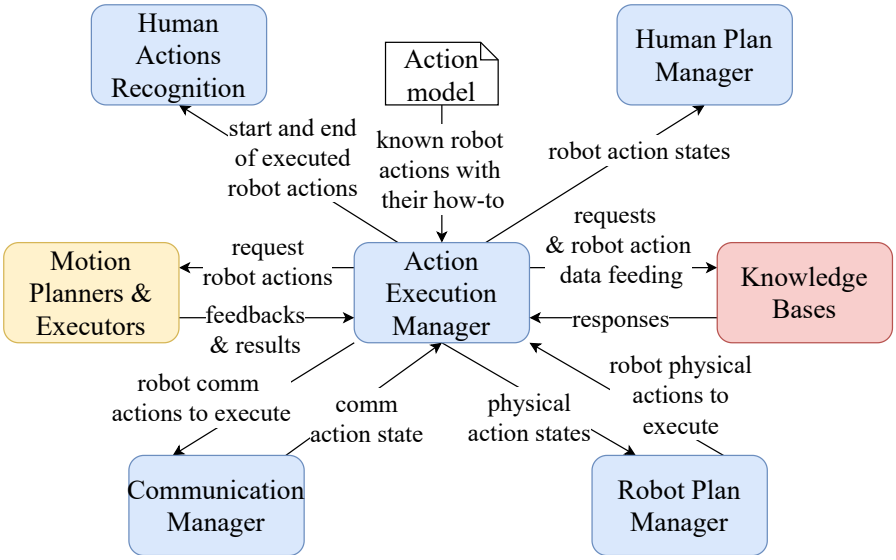


Figure 6.19: The Action Execution Manager and the RJAs (in blue) and the components of the robotic architecture (in red and yellow) with which it interacts. It is fed by a file (in white) with description of the robot actions the robot can do and how.

Deciding is not enough, the robot needs to be able to act. Thus, JAHRVIS as a ROS-Jason Agent (RJA) called the Action Execution Manager (AEM). Figure 6.19 shows its relations with the other RJA of JAHRVIS and components of the robotic architecture. The AEM is composed of a generic part managing the general flow of an action execution, described in Algorithm 7, and of a task-specific part, specifying the distinctive characteristic of given actions which are instantiations of the EXECUTE function in a separate ASL file as explained in Section 6.1.1. Moreover, all actions of the type PhysicalAction are realized based on action clients to communicate with the Motion Planners and Executors (see Figure 3.1) which allows a fine management of the execution through feedbacks and error codes. Finally, each instantiation of a PhysicalAction automatically starts and ends with the setting of the action parameters monitoring by the robot head, based on the head management presented in the next paragraph. For CommunicateAction, their parameters are reorganized and then the action is sent to the CM for execution.

Robot Resource Management The correct handling of the resources (head, arms, base...) of a robot is critical to perform a task, but it can be cumbersome for a deliberative component, such as JAHRVIS, to do all the micro-management required. To tackle this issue, a physical resource management system has been designed by Guillaume Sarthou and Guilhem Buisan, inspired by Devin [86]. For each of the identified resources is instantiated a component called *Resource Man-*

Algorithm 7 Action execution management

```
function ExecuteAction(Π_i) with agent_i ∈ Robot
    if name_i ∈ PhysicalAction then
        SendMessage([RPM,HPM,HAR], Π_i) with state_i = ONGOING
    else if name_i ∈ CommunicateAction then
                                          ▷ ONGOING state set by the CM
        SendMessage(CommManager,Π_i) with state_i = TODO
    action := ActionParsing(name_i,params_i)
    Execute(action)
    if name_i ∈ PhysicalAction then
        SendMessage([RPM,HPM,HAR], Π_i) with state_i = EXECUTED
    else if name ∈ CommunicateAction then
        SendMessage([RPM,HPM], Π_i) with state_i = EXECUTED
```

ager, having two types of input: permanent channels, that can be preempted at any time (*e.g.*, look at the head of the human interacting with it) and finite state machines which are not preemptable (*e.g.*, set of commands to scan a table). A component called *Resource Synchronizer* deals with actions requiring multiple resources such as the human-aware navigation which uses the head and the base. The synchronizer also reports the status of the ongoing coordination signal to JAHRVIS to monitor the progress of the action. Finally, a priority scheme has been implemented to handle multiple active inputs at the same time for one resource.

Such component allows JAHRVIS to be agnostic about the used robotic platform, as it offers the same interface for whichever one. Moreover, it enables joint attention with a nice head control as JAHRVIS can switch priorities between three types of permanent channels that have been defined, depending on where we are in the task: environment monitoring, human head monitoring and human hand monitoring. The two latter are set with the head and hand of the human the robot is interacting with as point to follow, whereas the environment monitoring channel receives new point of focus according to the needs of the task (*e.g.*, the cube the robot should pick or the box in which the human has to put an object). Thus, when the agents are talking together, JAHRVIS will set the human's head with the highest priority, but when the robot has to pick a cube, it will be the environment monitoring that will have priority. Finally, the head behavior can be controlled not only based on visual inputs but also on laser inputs for example if it has some. Indeed, according to the task context, it can be interesting for the robot to know what is going on around it. It this case, a channel can be instantiated so the robot can react when it detects moves with its laser and then looks in this direction.

6.6 Communication Management

The last RJA involved in the robot decision and control is the Communication Manager (CM). It is not dedicated to complex talks with the human but to enable the

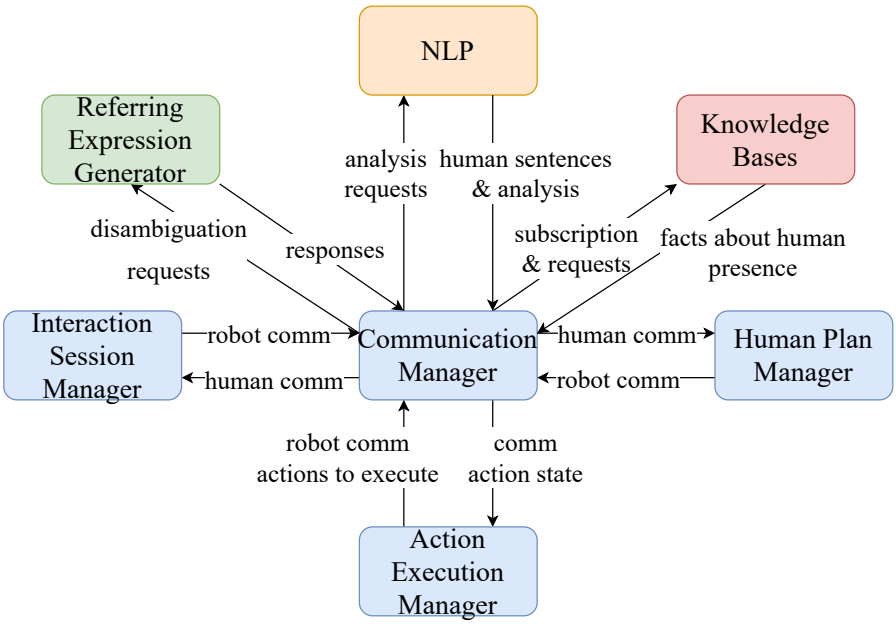


Figure 6.20: The Communication Manager and the RJAs (in blue) and the components of the robotic architecture (in red, green and orange) with which it interacts.

robot to make and understand communication during collaborative tasks, because as shown in Section 1.4, it is important. This process is based on a software for Natural Language Processing (NLP) and closely linked to a domain-specific planner called Referring Expression Generator (REG). This latter has been developed by Guillaume Sarthou and Guilhem Buisan [54]. It aims, regarding the current symbolic state of the world, at finding the minimal set of relations to communicate and allow the listener to identify a given entity. For example, if the robot wants to talk about a green cube on a table but there is another green one on a close shelf and a red cube on the table , how can it do? Well, it queries the REG which answers with a nominal group such as "the green cube on the table". Figure 6.20 shows its relations with the other RJA of JAHRVIS and components of the robotic architecture.

6.6.1 To Issue Communications

As shown above, three other processes, the Interaction Session Manager, the Human Plan Manager and Action Execution Manager, can query the CM to issue a communication to the human. We defined several types of communicative acts the robot can do:

- to do social chat for interaction session opening and closing (*e.g.*, "hello"),
- to give information about its abilities/role (*e.g.*, "I'm here to help you find your way"),
- to initiate a shared goal (*e.g.*, "Let's do this package"),

- to give information about the ongoing task (*e.g.*, "I cleaned the table"),
- to ask information about the ongoing task (*e.g.*, "Have you cleaned the table?"),
- to request the human to perform an action (*e.g.*, "Can you clean the table?"), and
- to give up the shared goal (*e.g.*, "I'm sorry I give up, I'm failing")

We focused on the communicative acts described in 6.6.1 and 6.6.1 with the verbalization of actions presented in Algorithm 8, developed in collaboration with Guillaume Sarthou.

As the communication is important, it is important to minimize the risk that it gets lost. Thus, when the CM receives a communication to perform from another RJA, it ensures that it perceives the human before issuing the communication so it has more chance to get the human's attention. Therefore, it sets the monitoring channel of the Robot Resource Manager on the human head monitoring (see Section 6.5).

Moreover, when the robot needs to communicate to the human that it has does a given action or to ask them to perform one, it needs to verbalize it properly. Still in the spirit of a generic system, the algorithm that we developed is not task or action specific. Action labels (presented in Section 6.1.1 and verb conjugation) are stored in Ontologenius which can be manually fed with new ones when needed.

Let's take an example where instead of making a stack, the agents have to remove the cubes from the table to put them in a green box. Then, we put ourselves in the case where the robot inform the human it has performed a Drop action `robot_drop_cube` in the plan, with `red_cube_1` and `green_box` as parameters.

First, the CM queries Ontologenius to get the closest class in the hierarchy with labels. Here, it is `DropAction`. There are three possible labels for this action, so the robot has the possibility to communicate about it with different action parameters: `[{Agent} @Drop {Cube}, {Agent} @Drop in {Container}, {Agent} @Drop {Cube} in {Container}]`. Thus, the CM finds which label matches its parameters, based on the parameter number and on their class. Then, in our case, this is `{Agent} @Drop {Cube} in {Container}`.

Next, for each parameter, it tries to find the right verbalization, either it will be a Referring Expression or the parameter label stored in Ontologenius, and replace them in the class action label. So, we would have something like `{Agent} @Drop the red cube on the table in the box`.

Finally, it replaces `{Agent}` with the action actor and `@Drop` with the right conjugation. As the CM wants to refer to an action the robot has done, it queries the Ontology with the conjugation of `Drop` at the past first singular person, which is `dropped`. Then, the result of the action verbalization is `I dropped the red cube on the table in the box`.

When the CM wants to ask the human to perform an action, it uses the same algorithm and turn the sentence into an interrogative form with the verb "can". As JAHRVIS manipulates conditional plans, the CM can take as input list of actions

and separate them with "or" when communicating about them. And, when it wants to communicate about the human having to perform several actions in a row, it using "and".

Algorithm 8 Action verbalization

```
function ActionVerbalization(agent,action,parameters,tense)
    labelList :=GetClassActionLabels(action)        ▷ Query to Ontologenius
    actionVerba :=LabelToWords(labelList,parameters)
    if agent ∈ robot then
        person := FirstSingularPersonalForm
        pronoun := I
    else if agent ∈ human then
        person := SecondSingularPersonalForm
        pronoun := you
    verb :=GetVerb(actionVerba)
    conjugatedVerb :=GetConjug(actionVerba,verb,person,tense)▷ Query to
                                                              Ontologenius
    actionVerba := actionVerba.Replace("{Agent}",pronoun)
    actionVerba := actionVerba.Replace("@verb",conjugatedVerb)

function LabelToWords(labelList,parameters)
    actionVerba :=""
    for each label in labelList do
        labelClassParams :=RegexMatch("{(?!Agent)(.*?)}",label)
        if Length(labelClassParams)=Length(parameters) then
            for each param in parameters do
                for each labelClassParam in labelClassParams do
                    if param ∈ labelClassParam then          ▷ Query to
                                                              Ontologenius
                        if actionVerba ="" then
                            actionVerba := label
                        if param ∈ Pickable, isAbove(param, Support) then
                            param :=GetReferringExpression(param)
                        else
                            paramVerba := GetParamVerba(param)
                                                    ▷ Query to Ontologenius
                        actionVerba
                                := actionVerba.Replace("{labelClassParam}",
                                                              paramVerba)
                        break
```

6.6.2 To Understand Communications

To have the robot enunciating information or asking some to the human is important but that it is able to receive communication from them is as well. The human should

be allow to communicate about the plan such as to ask precision about a given action, to signal that something is not going well or to ask the robot to perform an action. We focused on the latter in collaboration with Guillaume Sarthou.

The Algorithm 9 allows the CM to translate a human instruction about an action to perform by the robot into an instruction understandable by the Action Execution Manager.

Algorithm 9 Understanding of a human instruction

```
function GetActionToPerform(sentence,context)
    ⟨act, sparqlQ, score⟩ := GetSentenceSegmentation(sentence)   ▷ Query to REG
    if score > Score_min then
        merged :=MergeSparqlWithContext(sparqlQ, context)   ▷ Query to REG
        matchingObjects := SparqlToObjects(merged)   ▷ Query to Ontologenius
        if Length(matchingObjects)=1 then
            return act, matchingObjects[0]
        else
            question :="do you mean "
            for each object in matchingObjects do
                sparqlO :=GetSparql(object)   ▷ Query to Ontologenius
                objectVerba :=GetObjectVerbalization(sparqlO)   ▷ Query to REG
                question := question + objectVerba
            answer :=AskHuman(question)
            AnalyzeSentence(answer,merged)
```

Let's take an example where a green cube is on a table (`green_cube`) and another is on a shelf beside (`green_cube_3`). The human instructs the robot to take the green cube but there are two of them so we are going to see how the CM process to handle this order.

When the CM receives a human sentence such as "Take the green cube", it queries the Natural Language Processing (NPL) component which returns the action name it isolated from the rest of the sentence (*e.g.*, in our case, "take"), a SPARQL query corresponding to the parameter (*e.g.*, here, a SPARQL matching "green cube"), and a comprehension score (*e.g.*, with such a sentence it would be 1.0).

Then, it requests Ontologenius for the list of objects corresponding to the SPARQL query, from the human's perspective. Indeed, the robot could perceive a green cube which is not visible by the human (*e.g.*, in their back), in this case, it will not be part of the returned objects as it is not part of the human's knowledge. If the human has properly given their instruction, the object list size should be 1 and the algorithm is over (*e.g.*, in our example, if there was only one green cube).

However, for some reason, they may have been imprecise or absent minded (*e.g.*, here, they forgot that another green cube was on the shelf). In this case, the CM gets from Ontologenius the SPARQL query corresponding to each object, with elements allowing to discriminate between them. Then, it requests the REG for the verbalization of these objects, based on their SPARQL description. Thus, we could have the robot asking the human something like "Do you mean the green cube on the table or on the shelf?".

Then, the function starts over with the human answer which could be "the green cube on the table" – the CM keeps the action of the initial instruction into memory.

Finally, once the CM isolated an action and a parameter, it sends them to the Action Execution Manager, in our example it would be `take(pr2_robot,green_cube)`.

Chapter 7

Quality of Interaction Evaluation

Contents

The work presented in this section has been excerpted from a paper published in the Journal of Social Robotics [195].

7.1 Introduction

Robots dedicated to Human-Robot interactions are not just machines receiving commands and executing them. They should be decisional agents with high-level goals, taking decisions (potentially taking into account social norms), and acting and reacting to not only their actions but those of other agents. Cognitive and interactive robots are becoming more and more capable thanks to the use of human-aware models and algorithms [171, 285], with roboticists endowing them with the ability to execute their share of the work while adapting to contingencies, particularly those caused by human's behaviors and decisions [131, 14, 185]. The decision-making process is based on a range of knowledge about the environment, the interaction, the context... Nevertheless, curiously and interestingly, very little has been done to allow the robot, while performing its collaborative or assistive activity, to permanently evaluate if things are going well or not, as humans do. We name this ability "the measure of the Quality of Interaction from the robot point of view". We believe that enriching the robot knowledge with a good estimation about how the interaction is going, could enhance its decision-making process and thus, its social behaviour.

For example, if the robot detects that the QoI starts to drop, it can take a decision based on this information and act to try to improve the interaction quality

(*e.g.*, it can choose to change some modalities such as the language in which it communicates with the human, the volume of its speakers, or the parameters of its planners). On the contrary, when the QoI is high, the robot can decide to just continue the interaction as planned. Then, endowed with a QoI Evaluator, a robot becomes more adaptive and performs better. Also, a very poor performance all along a task could allow the robot to assess that the human is not really engaged in the interaction, or even is trying to play the robot. In such a situation, the robot might perhaps better disengage. Finally, from a methodological point of view, a robot deployed in the wild able to assess interactions, has an asset compared to others as it could reduce the investment in material and human resources to perform user studies. And, a developer might use the logs to improve their design.

In this paper, we only focus on the Quality of Interaction evaluation process and not on how to use its result for decision making. Therefore, we present in the sequel the methods and tools we developed, allowing the robot to evaluate in real-time the quality of the human-robot collaborative activity it is involved in. It is based on a set of metrics we have defined, focused on two concepts: the measure of human engagement and the measure of the effectiveness of collaborative tasks performance. However, this is by no means exhaustive, and other metrics and parameters could (and should) be added later. Our work can be seen as a toolbox among which it is possible to pick the desired metrics according to tasks or contexts. We propose a way to aggregate these metrics, producing the QoI. The evaluation of the QoI is performed at three different levels of abstraction: the interaction session level, the task level and the action level. In further work, this ability could provide additional information to the robot and open the possibility for reconsidering its behaviour in case it estimates that the quality of the interaction is degrading (*e.g.*, changing its plan or the way it is achieving it, informing the human or requesting a change in their behaviour, or even deciding to disengage).

The chapter is organized as follows. First, we briefly discuss related work and the main challenges. Then, in sections 7.3 and 7.4, we introduce our concept and proposed set of metrics to evaluate the Quality of Interaction. Example on a real task and proof-of-concept are presented in Chapter 8.

7.2 Related work

Inspired from the evaluation methods used in Human-Computer Interactions and User Experience fields, the field of Human-Robot Interaction (HRI) has elaborated its own methods to evaluate robotic systems when they interact with humans. There are various ways to evaluate a human-robot interaction from the human perspective. Bethel *et al.* [32] divided them into five categories: (1) self-assessments, (2) interviews, (3) behavioral measures, (4) psychophysiology measures, and (5) task performance metrics. They reviewed metrics used for each of the categories. They can be grouped into two types: (1) and (2) are subjective metrics and, (3), (4) and (5) are objective ones. Since our aim is to have a robot able to evaluate interactions

by itself, human subjective metrics are not usable. Then we focused on the study of existing objective metrics meant to measure how the interaction goes. Steinfeld *et al.* [278] proposed a set of metrics to be used in a wide range of tasks whose goal is to assess the system performance by measuring the task effectiveness (*i.e.*, how well the task is completed) and the task efficiency (*i.e.*, the time required to complete a task). Their work is very thorough and inspiring but does not target the evaluation of the quality of an on-going interaction. Hoffman [130] defined a type of quality of interaction, the *fluency*, pointing out that the notion is not well defined and somewhat vague but can still be assessed and recognized when compared to non-fluent scenario. To measure it, they proposed a list of objective metrics, only based on duration measures, designed to be quite general: robot idle time, human idle time, concurrent activity (*i.e.*, active time of both the robot and the human), functional delay (*i.e.*, time difference between the end of one agent's task and the beginning of the other agent's task). It is an interesting way to measure the fluency and thus the quality of the human-robot interaction but it only applies to shared workspace tasks and is dedicated to an offline evaluation.

Systems targeting real-time measurements during human-robot interactions, with the purpose to "close the loop" and use the information for decision-making, have been developed. Tanevska *et al.* [282] proposed a framework allowing the robot to perceive with face detection and evaluate in real-time the affective state (*i.e.*, anger, happiness, sadness, surprise, etc) and the engagement state (*i.e.*, whether the person is interested or bored in the interaction) of the people it is interacting with. However, the human affective state measure might not be enough to assess an interaction or a task as an affective state is actually a facial expression which can be misinterpreted (*e.g.*, a smile can be a sign of happiness or embarrassment) and which might be not visible when one of the agent performs an action and looks somewhere else. Moreover, as the notion of engagement is very task specific, it needs further exploration. Real-time engagement measurement has also been investigated by Anzalone *et al.* [8] using metrics such as gaze, head pose, body pose and response times. Their work is interesting and could be an element among others to assess the interaction quality but, it is dedicated to face-to-face interactions.

Cameras are not the only sensor used to assess interactions on-the-fly, some use human physiological responses such as skin conductance and temperature, heart or brain signals. Itoh *et al.* [142], Bekele *et al.* [26] or Kulic *et al.* [172] use them to detect human affective states such as anxiety or liking in real-time. However, physiological measures often imply a lot of sensors which can be invasive for the human. And, as explained by Kulic and Croft [173], physiological signals may be difficult to interpret and there is a large variability in physiological response from person to person. Thus, it can be difficult for a controller to determine which emotional state the subject is in, or whether the response was caused by an action of the system, or by an external stimulus. Moreover, we claim that the human affective state only is not enough to assess the quality of an interaction, a human could be satisfied with an interaction or a task result even though they were stressed during it.

Finally, Bensch *et al.* [30] proposed a formal approach to compute interaction quality in real-time. Their work focused on how to combine metrics together which is in the same line as ours. However, they do not provide implementation examples, remaining at an abstract level.

In summary, while a substantial number of studies have been devoted to the evaluation of collaborative interactions for analysis purposes once the interaction is over, there is a lack of methods allowing the robot to evaluate in real-time the quality of the interaction based on multiple metrics and not only anxiety or engagement. We claim that such an ability is very important and should strongly influence the situation assessment as well as the decisional abilities of interactive and collaborative robots.

7.3 The Quality of Interaction (QoI)

We believe the real-time assessment of the QoI with a human partner (*i.e.*, what the robot "thinks" about how the interaction is going) is a new knowledge that could enhance the robot decision-making process. We define the Quality of Interaction as a measure that indicates how good is the interaction during human-robot collaborative activities. It is computed in real-time based on a set of metrics, at three different levels: the interaction session level, the tasks level and the actions level. The QoI of a given level is computed from selected metrics but also from the QoIs of the level below as shown in Fig. 7.1.

The QoI of each level is computed as a score between [(1) for a good quality] and [(−1) for a poor one]. Metrics used to compute the QoI are divided in three categories:

- $Mp \in [0, 1]$ if it can only have a positive effect on the evaluation;
- $Mn \in [-1, 0]$ if a metric can only have a negative effect on the evaluation;
- $M \in [-1, 1]$ if a metric can have a positive or a negative effect.

Defined by the designer according to the needs and context, a metric can belong to one category or another depending on the target application. When needed, metrics values are scaled with the equations presented in Appendix B.

The evaluation of the Quality of Interaction at the level $l \in \{session_f, task_j, action_k\}$ (with f, j and k respectively the identifiers of a given interaction session, task and action), QoI_l, is computed with:

$$QoI_l = \frac{\sum_{i=0}^{x} W_i * M_i}{\sum_{i=0}^{x} W_i} + A * \frac{\sum_{i=0}^{y} Wn_i * Mn_i + \sum_{i=0}^{z} Wp_i * Mp_i}{\sum_{i=0}^{y} Wn_i + \sum_{i=0}^{z} Wp_i} \tag{7.1}$$

with W_i, Wp_i, Wn_i respectively the corresponding designer-set weights of M_i, Mp_i, Mn_i, A the designer-set weight of the right part of the + sign and x, y, z respectively the number of the metrics M_i, Mp_i, Mn_i.

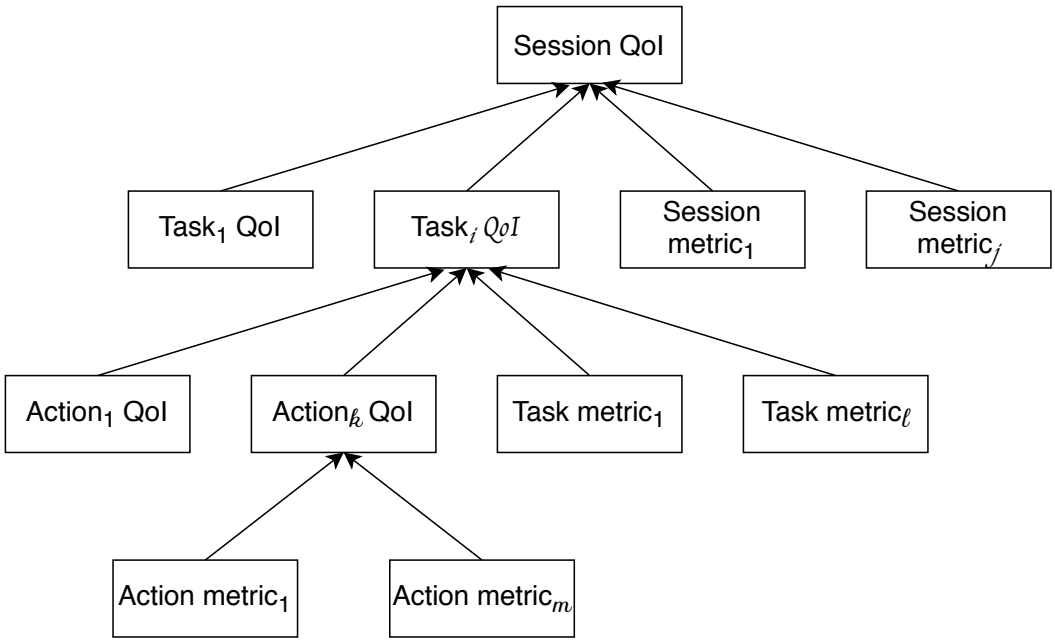


Figure 7.1: Representation of the QoI dependencies, with i the number of performed tasks during the interaction session, k the number of performed actions during the task i, j the number of metrics to measure the interaction session QoI, l the number of metrics to measure the task i QoI and m the number of metrics to measure the action k QoI.

Equation 7.1 aggregates the values of the metrics chosen to be indicators of the interaction level quality. As all metrics do not have the same importance in the measure of the QoI, each of them is weighted. Values of these weights are empirically defined. There are two parts in the equation, the left part of the + sign and the right part. The left part of the + sign is a weighted mean of the third category of metrics, the M metrics. The right part is a weighted mean of the metrics seen as bonus (*i.e.*, Mp metrics) or penalty (*i.e.*, Mn metrics). This latter part is weighted with A – whose value is also empirically[1] defined – to be able to adjust its influence on the left part. In such a way, if there are no Mn metrics to compensate for the Mp metrics, it is possible to limit the positive influence of the Mp metrics on the M metrics with A. It is the same if there are no Mp metrics, A can compensate the impact of the Mn metrics on the M metrics. Even though $M, Mp, Mn \in [-1, 1]$, the final result of QoI_l might be less than -1 or greater than 1 because of the addition of the M with the Mn and Mp. If it happens, QoI_l minimal value is set to -1 and its maximal value is set to 1.

[1]Values are empirically defined given intuition regarding the importance of a given metrics for a given task and a set of testing experiments

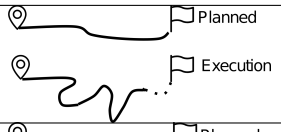
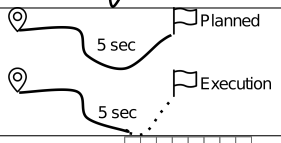
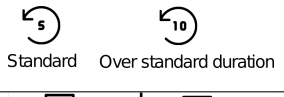
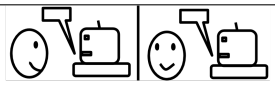
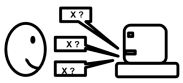
	Metric names		Measures	Illustration	Session	Task	Action
Effectiveness	Progress towards goal	Distance-to-Goal	Geometric distance			x	x
		Time-to-Goal	Time			x	x
		Steps-to-Goal	Number of executed actions/substasks			x	
	Deviation from standard duration		Time			x	x
Engagement	Fulfilling robot expectations about social interaction		*e.g.*, attention ratio, with-me-ness,...		x	x	x
	Human contribution to the goal		*e.g.*, number of repeated instructions, number of successful human actions,...			x	x

Table 7.1: The set of metrics presented in Section 7.4.

7.4 A set of metrics

In this section, we present a few measures to assess the QoI of an interaction session in Sect. 7.4.1. Then, we present metrics for the different levels based on engagement in Sect. 7.4.2 and effectiveness estimations during human-robot joint activities in Sect. 7.4.2. For example, if the human is engaged and if tasks are performed effectively, the QoI will tend to be high and *vice versa*. Both concepts are difficult to measure, so we do not exactly measure them but we compute their trends from the set of metrics presented in this section. This set is not exhaustive and will be extended in future work but it gave promising results as we show with our implementation in Chapter 8. All metrics are meant to be used for online evaluations of interactions. They are summarized in Table 7.1.

7.4.1 Measures to assess the QoI at the interaction session level

According to the context, the duration of an interaction session can be an indicator of the human engagement. Indeed, a human leaving only a few seconds after the beginning of the interaction is probably less engaged than a human staying with the robot several minutes. Also depending on the context, the number of executed tasks is a measure which can be considered as interesting information with respect to the engagement of the human, as well as the ratio of successful tasks. The more

the human executes successful tasks with the robot, the higher the session QoI might be. Finally, it can be valuable to take into account how the session has been terminated in the evaluation of the quality of an interaction session. For instance, the fact that the human leaves abruptly in the middle of a task, during an idle time or a conversation without saying goodbye, or only at an appropriate time saying farewell to the robot is significant in terms of social interaction quality.

7.4.2 Metrics related to human engagement

Michael *et al.* [199] stated that commitments[2] facilitate "the planning and coordination of joint actions involving multiple agents. Moreover, commitment also facilitates cooperation by making individuals willing to contribute to joint actions to which they would not be willing to contribute if they, and others, were not committed to doing so". As it is an important element of the joint action, we want to provide the robot with a way to estimate the engagement of its partner during an interaction.

Metrics allowing to state if an agent is engaged or not in an interaction are often specific to the type of interaction. For example, Fan *et al.* [93] implemented their measure of the human engagement as a kind of hysteresis: when the human gaze is on the robot, they are considered as engaged and when the human gaze is somewhere else during more than 3 consecutive seconds, they are considered as not-engaged.

In the same vein, we think that the measure of the engagement for a collaborative activity can be divided in 2 types of metrics, summed up in Table 7.1: the Human contribution to the goal and Fulfilling robot expectations about social interaction.

We define in this section examples of metrics of each types which can be used to estimate the level of engagement of the human partner.

Human contribution to the goal A good and very promising indicator could be the ability from the robot to evaluate how well the human actions help to the goal progression. We call this indicator *Human contribution to the goal*. To the best of our knowledge, there is no general method to estimate it.

As a first version of the *Human contribution to the goal*, we chose to measure it through the number of times the robot has to repeat an instruction or a question before the human performs correctly, when it expects the human to answer or to perform the action. As, if it needs to repeat, it means that the human is not correctly contributing to the goal, intentionally or not, as they are not performing their part of the HR action as they should. The more the robot needs to repeat because of the human's bad performance, the less they are contributing to the goal, the more the action QoI should decrease.

[2] In the robotic domain, it is the word "engagement" and not "commitment" which is often used, unlike in the psychological and philosophical fields.

Fulfilling robot expectations about social interaction During a social interaction, agents are expected to behave in a certain way and so the robot has expectations about the human. Then, the robot can monitor the human behavior to check if they are acting as they are expected to. For example, most of the time, when the robot speaks to the human, it will expect them to look at it and so it can monitor if it is the case or not as implemented by Fan *et al.* [93]. Quite similarly, Lemaignan *et al.* [183] developed a way to measure if the human is *with* the robot during their interaction, based on attention assessment, by computing if the human is looking at the desired attentional target or not. This latter metric will be integrated to our framework in future work.

As the works of Lemaignan *et al.* and Fan *et al.*, we estimate the *Fulfilling robot expectations about social interaction* with the human head orientation, in the context of our implementation described in Chapter 8. We compute an attention ratio *i.e.*, the time during which the human is attentive to the robot (*i.e.*, staying close enough and looking at it) when it speaks compared to the total time of the speech:

$$Ar = \frac{duration_{isAttentiveTo(robot)=true}}{duration_{robot\_speaks}} \tag{7.2}$$

Metrics related to effectiveness One can elaborate metrics to measure how well a task or an action is achieved. As discussed by Olsen and Goodrich [214], there are a variety of metrics such as time-based metrics which reward the speed of performance or the response times; error metrics which are based on counting retrials, failures, or mistakes; coverage metrics which measure to what extent a goal is achieved, as well as other possible metrics. We use some of them such as counting retrials, however these metrics alone were not enough for our example task as we are in an HRI context.

One can measure for different kinds of tasks, the ratio of successful[3] executions to the total number of executions (*e.g.*, $R = \frac{Succ}{Exec}$) or the deviation from the initial plan (distance, cost, trajectory, etc).

We define four metrics, summed up in Table 7.1, allowing to measure the current task and action effectiveness. Three of them are means to measure how the progress towards the goal of a task or an action varies. Indeed, they are good indicators for the interaction quality as, when executing a task or an action, if the agents are not getting closer from the goal or even diverged from it, it means that something goes wrong. There are three different metrics because the one to use depends on the type of task or action. The fourth metric allows to compare the current execution duration to the standard execution duration of the task or action, based on durations measured during previous executions.

[3]Obviously, the success is context and task dependent and should be defined according to the needs

Metrics to assess the progress towards the goal We defined three different metrics to assess the progress towards the goal. The first one allows to assess the progress towards the goal of geometric-based actions. The second estimates the progress by using the remaining time to reach the goal. Finally, the last one measures the number of remaining steps (actions or substasks) before achieving the goal of a task.

Distance-to-Goal When an agent is performing a geometric-based action such as a movement, observing if the agent is getting closer to the target position over time provides a useful information about how well the action is going. Therefore, we introduce the *Distance-to-Goal* ΔDtG metric:

$$\begin{cases} \Delta DtG(t=0) = 0 \\ \Delta DtG(t) = \max(0, \Delta DtG(t-1) - 1) \\ \qquad\qquad \text{if } path\_length(t) < path\_length(t-1) \\ \Delta DtG(t) = \Delta DtG(t-1) + 1, \text{otherwise.} \end{cases} \tag{7.3}$$

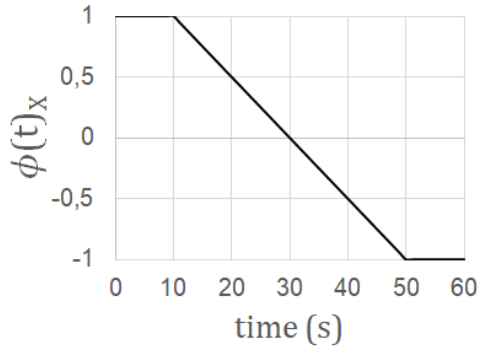
with $path\_length(t)$ the length of the path leading the goal at time t (*e.g.*, which can be given by a reactive motion planner [159]). The metric lower bound is 0. If at time t the agent is closer to its final position than at $t-1$, *i.e.*, progressing towards their goal, the metric is set to decrease or to remain equal to 0. Now, if the agent has not moved or is even further, the metric increases. The closer the metric value is to 0, the better it is, as it means the distance to the goal has decreased over time. We chose to not directly compute the difference between $path\_length(t)$ and $path\_length(t-1)$ as the results would be very different whether it is an action implying a long path or a short path.

Time-to-Goal This measure is intended to estimate the progress of a given task or action towards its goal based on the estimation of the remaining time to reach it. It compares the current estimated time to goal with the initial estimated time to goal taking into account the current task duration. As so, it is possible to measure the variation compared to the initial plan. We define the *Time-to-Goal* ΔTtG as:

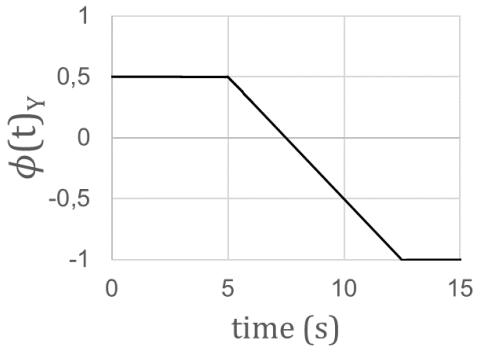
$$\Delta TtG(t) = \max(0, e(t) + TtG(t) - TtG(T_0)) \tag{7.4}$$

with $e(t) = t - T_0$ the task execution duration (time elapsed since the beginning of the task), $TtG(t)$ the current time to the goal, and $TtG(T_0)$ the initial planned time to goal. In our work, $TtG(t)$ and $TtG(T_0)$ are provided by a reactive motion planner [159] because we used the metric for navigation but it could be provided by other kind of planners.

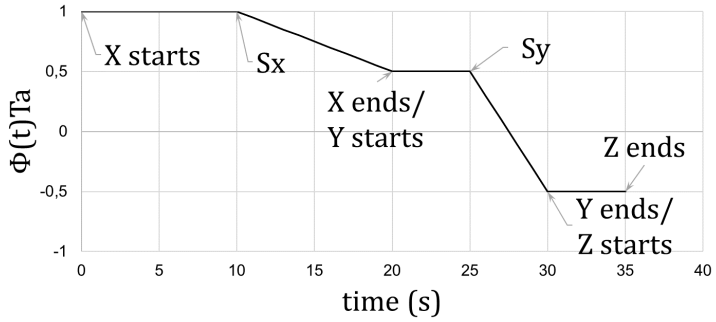
Steps-to-Goal One way to estimate the remaining distance to the goal for a task is to count the number of remaining substasks or actions (depending on the relevant scale) to perform. In addition, one can add a factor which estimates the



(a) Plot of $\phi(t)_X$ of the subtask X lasting 60 seconds, with $SD_X = 10sec$, $V_X = 0.5$ and $\alpha = 1$



(b) Plot of $\phi(t)_Y$ of the subtask Y lasting 15 seconds, with $SD_Y = 5sec$, $V_Y = 1$ and $\alpha = 0.5$



(c) Plot of $\Phi(t)_{Ta}$ for a task composed of a sequence of three subtasks X, Y, Z: the duration of X exceeded $SD_X = 10s$ and reached $20s$, the duration of Y exceeded $SD_Y = 5s$ and reached $10s$, finally the duration of Z was less than $SD_Z = 10s$

Figure 7.2: Examples of plots of the ϕ and Φ functions

weight (or effort needed) of each action or subtask. These weights can be determined by the designer, provided by the planner, etc. Then, the *Steps-to-Goal* $\mathcal{D}$ of a task can be computed as time t:

$$\mathcal{D}(t) = \frac{\sum_{i=1}^{c} \mathcal{W}_i}{\sum_{i=1}^{n} \mathcal{W}_i} \tag{7.5}$$

with $\mathcal{W}_i$ the weight of a subtask/action i, c the number of completed subtasks/actions and n the total number of planned subtasks/actions.

Deviation from standard duration We introduce here a metric to measure the deviation from standard execution duration, the *Deviation from standard duration* ϕ for subtasks/actions and the *Deviation from standard duration* Φ for a whole task. This measure is intended to represent the degradation of the quality of execution of an HR task when its duration exceeds a certain time.

To each subtask/action a_i, we associate two attributes whose values are defined by the designer: a soft deadline SD_i and a decreasing quality speed V_i. If, at time t, the execution duration $e(t) = t - T_0$ of a substask or action a_i which has started at T_0 exceeds SD_i, the quality will decrease over time at speed V_i:

$$\phi(t)_i = \max\left(V_i * \frac{-\max(e(t) - SD_i, 0)}{SD_i} + \alpha, -1\right) \tag{7.6}$$

where α is the value initial value and the upper bound (as at $t = 0$, $\max(e(t) - SD_i, 0) = 0$) of ϕ_i, when the subtask/action a_i starts.

Then, we define a metric Φ for a task. It is an aggregation of the ϕ_i computed for each performed subtask/action a_i of the task. At any moment, Φ can be seen as a memory of the previous steps, so the initial value α of a_i is equal to the final value of ϕ_{i-1} of the previous subtask/action a_{i-1}, $\alpha = \phi(T_{final})_{i-1}$.

We can notice that it is not possible for this metric to increase over time since it memorizes the values of the previous actions. However, the total computed QoI can get higher thanks to the other metrics. Moreover, ϕ can be used independently of Φ. In such a case, the initial of value α of ϕ can be set to 1.

Three examples are given in Fig. 7.2. Fig. 7.2a and 7.2b represent $\phi(t)_X$ and $\phi(t)_Y$ for two independent subtasks X and Y. Fig. 7.2c is a plot of $\Phi(t)_{Ta}$ for the task Ta composed of the subtasks X, Y, Z with $SD_X = 10s$, $V_X = 0.5$, $SD_Y = 5s$, $V_Y = 1$, $SD_Z = 10s$ and $V_Z = 1$.

7.5 Conclusion

In the last section, we have described a novel concept, the evaluation of the Quality of Interaction (QoI). This concept will be demonstrated in Section 8.7, in the context of the direction-giving task presented in Chapter 8.

change according to how we'll modify the plan

Part IV

Deploying and Evaluating an Interactive Robot

Chapter 8

A direction-giving robot in a mall

Contents

This chapter is from an article submitted to the User Modeling and User-Adapted Interaction (UMUAI) Journal. This work has been achieved in collaboration with Guillaume Sarthou, Guilhem Buisan, Phani-Teja Singamaneni, Yoan Sallami, Kathleen Belhassein, and Jules Waldhart. In this chapter, we first give an

overview of the European H2020 Project MultiModal Mall Entertainment Robot (MuMMER)[1]. We then present the components developed by the LAAS-RIS team. My technical contributions are related to the Supervisor component, the component integration, the overall system debugging, the real-world deployment and the user study.

8.1 Introduction

In large scale indoor environments, like museums, shopping malls, or airports, the presence of large interactive screens, maps, or signs underline the importance of providing information on itineraries. However, orienting and reading maps to find one's own way may be challenging. As for signs, the wanted written information may not be within sight. People also look for information not available on visual media such as the location of a given product. That is where the robot has a role to play, bringing a new way to help people to get their bearings in large indoor environments such as shopping malls.

Therefore, in the context of the European H2020 Project MuMMER[2], we developed and deployed a social service robot in one of the largest malls of Finland, Ideapark in the city of Lempäälä. This social robot is able to engage, chat with people, and guide them. We will not talk about the two first mentioned behaviors, developed by our project partners, but focus in this paper on the direction-giving.

As the mall has approximately 1.2 kilometers of shopping and pedestrian streets and more than 150 shops, people get easily lost. In such a large environment, having a robot guiding customers to their wanted destination would be time-consuming for the robot and would prevent this resource to be available for as many customers as possible. Inspired by the manner in which the mall employees perform this activity, we chose the solution to have a robot not accompanying people to their desired destination but rather verbally describing the route while grounding it with pointing gestures. If necessary, it moves a few meters inside its dedicated area (Figure 8.14) to improve the perspective sharing with the human when pointing at a landmark, and therefore to improve the human understanding of the route. These features are unique to a robot and cannot be found on a map or an interactive screen. To endow the robot with such abilities, we built a complete implementation of a robotic architecture that has been deployed in a real-world environment, the Finnish mall. There, it ran for three months, three days a week. Here is a sum-up of the project steps:

1. March 2018: beginning of the design and implementation of the direction-giving task

2. September 2018: First tests of the task on the field, *i.e.*, in a Finnish mall

[1] `http://mummer-project.eu/`
[2] http://mummer-project.eu/

3. June 2019 and September 2019: New tests of the direction-giving task on the field

4. From September to December 2019 (project formal end): The robot autonomously ran three days a week in the mall (with only remote monitoring of the robot performance by our team for debugging and tuning)

   (a) November 2019: Integration in the *Supervisor* of a preliminary version of Quality of Interaction Evaluator implementing the model described in Chapter 5
   $\Longrightarrow$ version 1 of the QoI Evaluator

   (b) From November 2019 to December 2019: Around 350 direction-giving tasks were performed with usual mall customers. Bug corrections and tuning of the direction-giving task. This allowed us to improve the QoI Evaluator thanks to: (1) data collection of task failures and standard durations of the subtasks executions (2) lessons drawn about metric definitions and choices.
   $\Longrightarrow$ version 2 of the QoI Evaluator

5. January 2020: User study with 35 participants to compare three direction-giving task robot behaviors, allowing to log interactions at the same time we could monitor them[3]. End of the project.

6. March 2020: Refinement of the QoI Evaluator, *i.e.*, improvement of the metric functions and tuning of their parameters. In the lab, with the same direction-giving task than the one used in the mall, comparison of the QoI computed by the robot when it is dealing with an "ideal" human, a "confused" human and a "non-compliant" human.
   $\Longrightarrow$ version 3 of the QoI Evaluator

All along the process, we elaborated and built the system based on the main principles and ingredients which have been identified and are investigated by the Human-Human Joint Action community. We also conducted preliminary studies and used the Joint Action perspective to analyze how human guides would achieve such an activity at the place where the robot was intended to be deployed. This was possible essentially because we were able to combine the results of the JointAction4HRI[4] project with the MuMMER project.

Our claim is that such an approach is relevant in the way the joint action principles provide pertinent guidelines and it is possible to effectively elaborate models and implement systems based on them. The output is a complete robot architecture that integrates a number of components implementing the main decisions and behaviors which have been identified. Each of them makes use of various models

[3]The QoI Evaluator was running in background, it was not the purpose of the study.

[4]It is a multi-disciplinary project which gathers philosophers, developmental psychologists and roboticists. `https://jointaction4hri.laas.fr/`

and decisional algorithms, all integrating explicitly human models and joint action principles and mechanisms.

The chapter is constructed as follows. In Section 8.2 we provide background information about robot guides and direction-giving task and discuss about how the human partner has been considered. In Section 8.3 we discuss how we model the direction-giving task as a human-robot joint action. We analyse the task based on human-human exploratory studies and decompose it into a succession of precise subtasks in Section8.4. An overview of the resulting architecture and a description of its components are presented in Section 8.5. Then, we present in Section 8.6, the integration of the overall architecture into a physical robot and the steps until its final deployment "into the wild". In Section 8.7, we show how we used this task to implement the QoI Evaluator presented in Chapter 5. Finally, we present the user study we performed with 35 participants and its results.

8.2 Related work

A number of contributions have proposed robot guides, from the first museum guides [55, 286, 267, 76] to more recent robot guides in large areas [21, 293]. For example, Chen *et al.* presented a guiding robot in a shopping mall where it accompanied the customer to the desired location and pointed at the shop [69]. Another example is a shopping robot helping people to find products among the aisles of a store [117]. However, the focus in these contributions is mainly the fact that the robot is challenged to navigate until the goal destination with the presence of humans. Efficient mapping and localisation in large areas, social navigation are the main concerns. This is different from our needs where the robot is voluntarily constrained for its motion to a limited area with a focus on conveying to the human the pertinent information to reach by herself the desired place.

Direction-giving tasks have been investigated in the human-robot interaction community. Kopp *et al.* describes an embodied conversational agent giving route directions using deictic gestures [167]. A number of key contributions have been developed over the years by ATR-IRC within the Robovie robot and project. First, Okuno *et al.* developed a model for a robot providing route directions, integrating utterances, gestures, and timing [213]. The experiments explored the influence of gestures and highlighted the importance of timing in the directions-giving task. Then, Kanda and colleagues implemented a guiding behavior as part of a wider system with the robot pointing toward the first direction to take and saying "please go that way" and then, continuing its explanation by saying "After that, you will see the shop on your right." [148, 149]. Their robot also gave recommendations for restaurants and shops based on customer tastes. In their following work, they presented a route perspective model attempting to represent humans' concept of route and visibility of landmarks, which they believed to match people's perception of the environment [204]. Then, Matsumoto *et al.* developed a robot able to follow a user while inferring their memory recall of shops in the visited route [193].

When the user asked the location of other shops, it gave the route description with references to the known locations inferred with the model of the user's memory recall. Finally, Satake *et al.* showed a complete architecture of an information-providing robot able to move around a square in a mall composed of: a map, an ontology, a speech recognition system (operated), a dialog manager, a localization module, and a people tracker. As in their previous works, the robot verbalized utterances and used deictic gestures to give route directions [247].

Let us also mention the work of Bohus *et al.*, a robot providing verbal directions to people using deictic gestures coupled with spoken references [34]. For example, the robot said "Go to the end of this hallway", executing a pointing gesture at the same time, and then continued the explanation with sentences such as "Turn right and keep walking down the hallway". Iocchi *et al.* mentioned both guiding and direction providing as use cases of their system [141].

Numerous other contributions can be found but, only a few of them propose full architectures for an autonomous direction-providing robot, the most complete one being the Robovie robot presented above.

Still, to the best of our knowledge, no system tackles the overall guiding-task with flexibility. Indeed we claim that it is important for the robot to reason about the current and desired perspectives of the human and the robot and to be able to pro-actively propose to the human a pertinent placement. This is one of the basic bricks of our system and it is strongly linked to the key principles of Joint Action which involve the ability to establish and monitor joint attention, and to conduct a multi-step task achievement involving contributions of both agents. Besides, it is the duty of the robot to permanently adapt to human needs and preferences and to synthesize acceptable behaviours.

8.3 Rationale

In Section 1.3, we presented the concepts around joint action. In this section, we bring some new inputs specific to the direction-giving task.

The design of our system has taken into account the results of several user studies involving human guides of the mall (see Section 8.4). Indeed, it could be of interest to have a robot performing in the same way as a human guide does. "If robots could display predictable behaviours that are in line with human's expectations based on their models of human joint action, the resulting interaction would achieve greater naturalness" [82, p. 17] and "human agents would then be able to apply predictive and adaptive processes acquired in human interactions to the interaction with robots" [82, p.17]. In the context of the direction-giving task with a Pepper robot, we take advantage of the fact that the robot is a humanoid and the human anthropomorphizes the robot behaviour (whatever we do). However, it is not always possible or desirable for a robot to imitate what a human would do at its place. It could let people think that the robot has more capabilities than it really has. In that way, besides the imitation, it could be desirable for the robot to exhibit its

limitations, *e.g.*, saying that it is able to provide you direction into the mall (and nothing else).

In our task, the robot has a role, it is a guide and the human is a customer with a need to find a direction. The joint action is not symmetric, there is a difference of knowledge and skills between the two agents. Curioni *et al.* raises the point that "task asymmetry is an important factor to consider when investigating complex joint action settings because it drives the systemic emergence of communication and coordination dynamics (for example in the form of task distribution)" [82, p. 11]. At the supervision level (see Section 8.5.7), we modeled which part of the task falls to the robot and which part of the task falls to the human. We can also infer that knowing the robot role as guide, the human would be able to infer what it is entitled to do. This way, we consider that they share the route description task representation. Another important point is that "shared task representations not only specify in advance the individual parts each agent is going to perform but they also govern monitoring and prediction processes that enable interpersonal coordination in real time." [165, p. 65]. Our system handles that monitoring and prediction in its supervision component(see Section 8.5.7).

And, in our system, the situation assessment component provides visual perspective-taking. It computes, from the robot point of view, a number of facts regarding what the robot is looking at, which landmark is visible to it, what is present at its proximity, etc. It also computes the same information from the perspective of the person interacting with it. This way the robot is able to infer, based on its own models, which information is shared (or not) with the person it interacts with.

8.4 Designing direction-giving behavior in a shopping mall

8.4.1 What we learnt from humans

In order to inform the design and implementation of the pertinent functions and their articulation, two human-human exploratory studies were conducted in collaboration with VTT Technical Research Centre of Finland. It allowed us, in addition to the study of the existing literature, to enrich our knowledge on effective route descriptions and how they can be used in the very context of the actual robot deployment environment.

The first pilot study consisted in a human guide providing route information. It was carried out close to the future location of the robot in order to avoid biases linked to the location or the environment. Based on preliminary interviews with guides working at Ideapark, a list of 15 shops often requested by customers was selected. The preliminary experiment consisted of one participant asking for shop directions to a guide working at the mall information booth. Two researchers, as participants, and two guides took part in the experiment. The two guides were instructed to give guidance as they would normally do. The situations were video

Figure 8.1: Picture from the second Human-Human study [27]. Here, the guide is giving the route description to reach a given shop by pointing at it. Positions regarding the target and the customer, as well as gazes and pointing gestures, were analyzed.

recorded and the guides were briefly interviewed after the sessions. The video analysis focused on non-verbal communication, and in particular the different types of gestures used to give guidance, the positions of the two protagonists in relation to the target shop and their interlocutor, and the gazes alternation. Belhassein *et al.* gave the first indications to consider for the robot guidance to be effective and understood by customers, resulting from this pilot study [27]. For example, this pilot study allowed us to notice a preferential use of the ipsilateral hand to the visual field of the target. In line with the existing literature on gestures studies, we also noticed that deictic gestures were naturally more frequent than iconic gestures or beats, while metaphorical gestures were rare. As shown by Allen, the hand used to point a referent was oriented vertically in the case of stores (vertical referents) or directed actions such as a path to take or turns, whereas in the case of horizontal referents (*e.g.*, escalators), the hand was oriented horizontally (palm facing the ground) [5].

A second exploratory study was then carried out adding complex situations (*e.g.*, two customers requesting directions at the same time, two different shops in the same request, or someone who interrupts the conversation between the guide and the customer). Again, social signals were analyzed (see Figure 8.1 for an example). The protocol used and the results have been published [125, 126]. By analyzing the sequencing of the whole interaction, this second study showed the guide pointing the general location of the target first, before explaining and pointing the different stages of the path to take to get there. Then, the sequencing of the route description

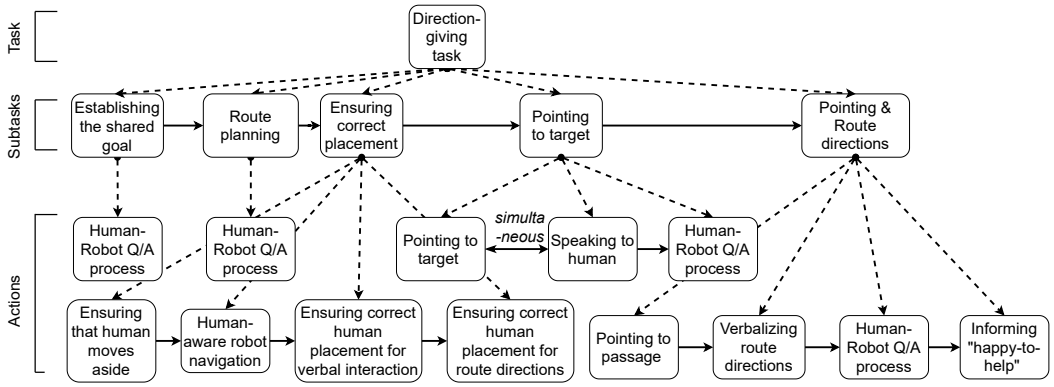


Figure 8.2: The representation of the direction-giving task as a hierarchical task network with task, substasks and actions levels. All the horizontal arrows are sequential links and the rest are decomposition ones.

itself showed that a first deictic gesture on a visible passage (corridor, or if the shop requested is on the second floor, the escalator) preceded the explanations about the directions to take. The most interesting results concerned situations of confusion and misunderstandings. Indeed, several elements might be sources of confusion for the customer, such as using only one transmission channel (*e.g.*, gesture without speech), the choice of landmarks which are not always appropriate, if there are several route descriptions in the same explanation, or when the distance is not specified.

8.4.2 Design of the collaborative task for a direction-giving robot

From the analysis of human-human direction-giving and through an iterative design process, we designed and implemented our directions providing robot. Our model of the collaborative task can be represented as a succession of subtasks, as shown in Figure 8.2. This figure also exhibits the incremental refinement of the task into a sequence of human-robot interactive actions. The aforementioned subtasks are:

1. **Establishing the shared goal**: In this first step, the human and the robot negotiate and establish a shared goal. Specifically, the robot tries to determine precisely the place – we called it the *target* – it should give directions for. This is immediately completed if the human directly asked for a known shop. Several verbal exchanges can be necessary in case the person asked for a kind of shop (*e.g.*, restaurant) or a product or in case the robot has not properly understood the name of the place and needs to disambiguate.

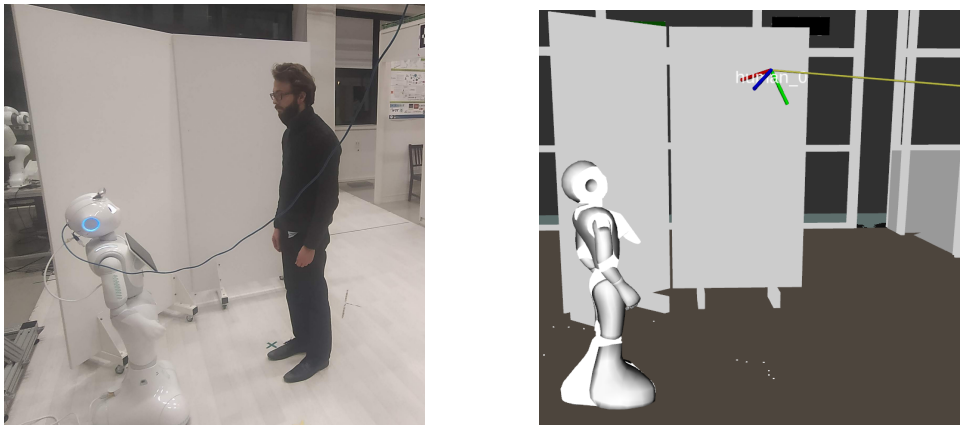
2. **Route planning according to the human willingness and ability to climb stairs**: As the robot role is to help people, adapting to them, it needs to ensure that they have the abilities to follow the route it will indicate to them. So, first the robot computes the best route to the target and then checks the presence of stairs in it. In case there are, the robot enquires whether the

human can or want to climb them or not. If they cannot or do not want to, the robot computes a new route without any stairs. The planned route contains a first *passage* (*i.e.*, a corridor, a door or an escalator) which the robot will try to point.

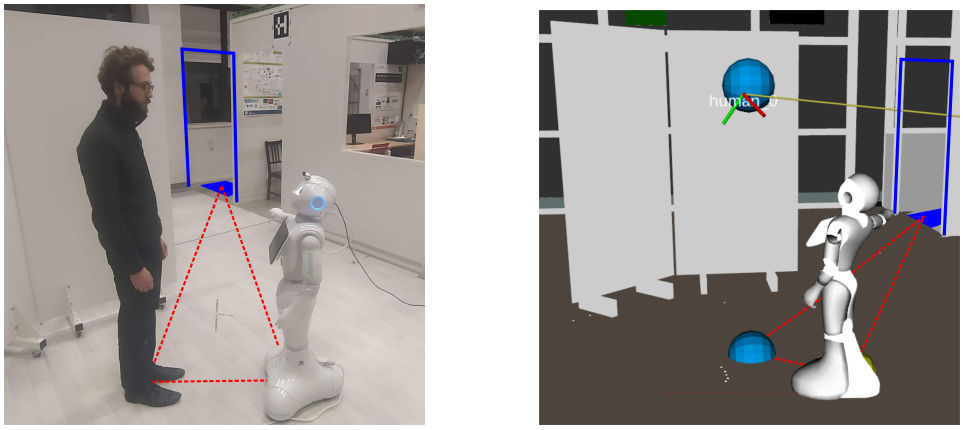
3. **Ensuring correct placement**: The second human-human study, mentioned in Section 8.4.1, highlighted the fact that human guides point to a visible *passage* before giving the route directions. Thus, we endowed the robot with this ability as described further in the item 5 of this list. In order to be in good conditions while performing this item 5, that is to say to ease the human understanding of the directions, the robot seeks better positions for the human and itself. It does so by computing a position for the human, considering their visual perspective of the passage. The robot computes a new position for itself as well, to form a triangle whose vertices are the planned robot position, the planned human position and the passage, as shown in Figure 8.1 and Figure 8.3. After having computed these positions, the robot moves, and as they both are engaged in the task, expects the human to join it once its position is reached; it calls them if they do not. As the human might not be at the exact position computed for them, the robot checks their visibility of the passage. In case their visibility is too low, the robot will adjust their position thanks to verbal instructions (*i.e.*, come closer, move back). Figure 8.3 illustrates the initial and final positions of both agents, in a lab context.

4. **Pointing to target**: Following the sequencing obtained from the aforementioned human-human study, the robot first points in the target direction, along with a brief sentence. As the robot is a helper and it is involved in a joint action with the human, it needs to ensure that its actions produce their expected results. In this case, if the robot computed that the target should be visible from their position, it checks that the human has seen it, either by monitoring their perspective or by asking. In case of a negative answer, it will point again.

5. **Pointing to passage and giving route directions**: Still following the sequencing from the study, when the target is not in the same physical space as them, meaning that there is a passage on the way to the target, the robot points to this passage and then verbalizes the route directions. These directions take into account the orientation the human will have and describe the route (*e.g.*, take the corridor on the left side). Finally, the way they are built (*i.e.*, the order of the steps, the keywords to use...) is also based on the human-human study. Here again, the robot ensures that the route directions have been understood by asking the person about it or if the passage has been seen if there is one. In case of a negative answer, it will point and give the route directions again. Finally, the robot ends the task with a "happy-to-help" short sentence.



(a) Initial positions of the human and the robot. The human asked the robot for route directions to a target behind him.



(b) The robot and the human are in their final positions. The blue spheres are the computed position for the human by the robot. The robot is pointing to the passage (in blue frame). We can observe the triangle formed between the human, the robot and the passage (the blue area on the floor) as in Figure 8.1 where two humans are in a triangle formation.

Figure 8.3: Initial and final positions of a direction-giving task in the lab context. On the left are pictures and on the right screenshots of Rviz[5](a 3D visualization tool for ROS.)

To endow a robot with the abilities described above, to build a robotic architecture embedding all these aspects, is a challenge. We tackled it with the architecture presented in the next section.

8.5 The deliberative architecture

est-ce qu'il faut charcuter des détails ?

In this section, we present the robotic architecture developed to handle the direction-giving task. It can be seen as an instantiation of the architecture presented in Section 3.3.

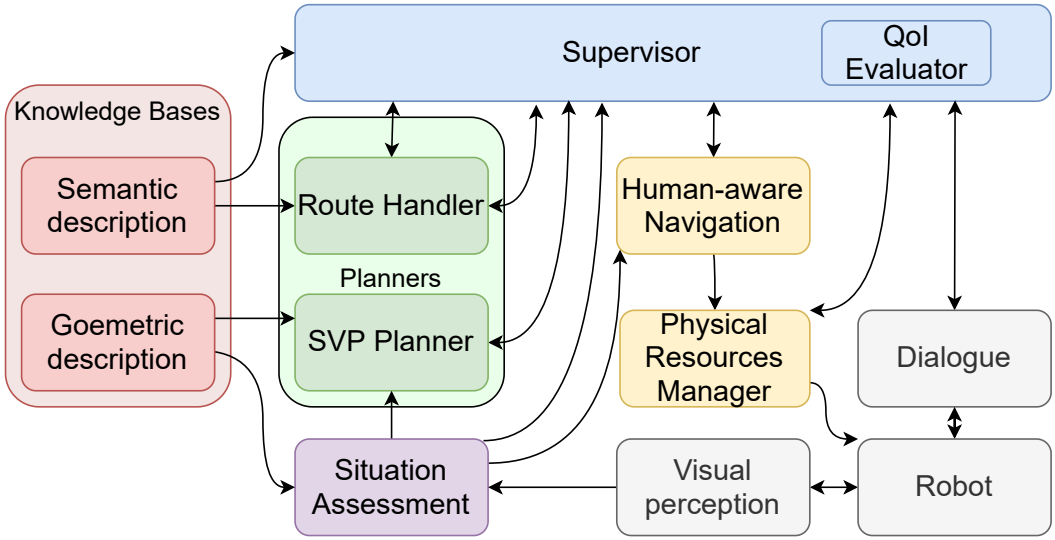


Figure 8.4: The general architecture of the system. The components presented in this paper are in[193] white. The visual perception and dialogue components have been respectively developed by IDIAP and HWU and are described in [99]. Naoqi is a Softbank Robotics software.

The figure 8.4 represents the architecture, its components, and their interconnections. Communication between components relies on ROS. In this chapter, we only present the components developed by the LAAS-RIS team, represented by the colored blocks on the architecture. First, we present the two knowledge representations in the form of geometric and semantic representations. Next, we introduce the components related to the sensorimotor layer. It is the situation assessment and the physical resource manager. Then, we present the components related to the deliberative layer. They are the Human-Aware Navigation, the SVP (Shared Visual Perspective) planner, the Route Handler, and among the key components, we finish with the supervision and control system, designed to operate human-robot joint tasks in a joint action context.

8.5.1 Environment representation

For a service robot providing directions to people, we need information to understand humans' need, information to compute the route to the goal, and information to compute the visibility of both agents to plan the pointing position. To understand the needs of a human wanted to be guided, we need information about the type of stores and the sold items. To provide so, Satake *et al.* used an ontology [246, 247]. To compute the route to the final destination, some works show the use a topological map [193, 213]. Each node of the graph is related to a 2D position of the environment. To estimate the human visibility of elements anywhere in the environment, Matsumoto *et al.* used a simplified 3D model where shops are represented by 3D polygons [193]. In our implementation, we only used two types

of representation of the environment: a **geometric** and a **semantic**.

Since the final deployment of the robot was in a Finland mall, we have built an mockup mall in our lab for development purposes. By mockup, we mean that shops signs have been displayed in the laboratory to create configuration similar to the real mall. The representations describe hereafter have thus been created both for the real mall and the mockup one.

8.5.1.1 Geometric representation

The geometric representation is used to compute the visibility of elements of the environment from different positions needed for the pointing of landmarks. However, because the robot does not accompany the person to the final destination and therefore does not move much, the possible visibility of the two agents is limited to their immediate environment. For this reason and due to the large scale of the Finland mall, we chose to geometrically describe only the subpart of the global environment that could be visible from the interaction area. For the rest of the environment, we represented the shops with 3D points only. These points are enough to point in the right direction. The resulting geometrical representation is a three-dimensional mesh model, as shown in figure. 8.5a for the mockup mall and in figure 8.5b for the real one. We have represented in the 3D model all the elements that could hinder visibility, such as poles or panels. In this way, we can precisely emulate human visibility. The model was created from the architectural plans first and then refined with measurements in the mall.

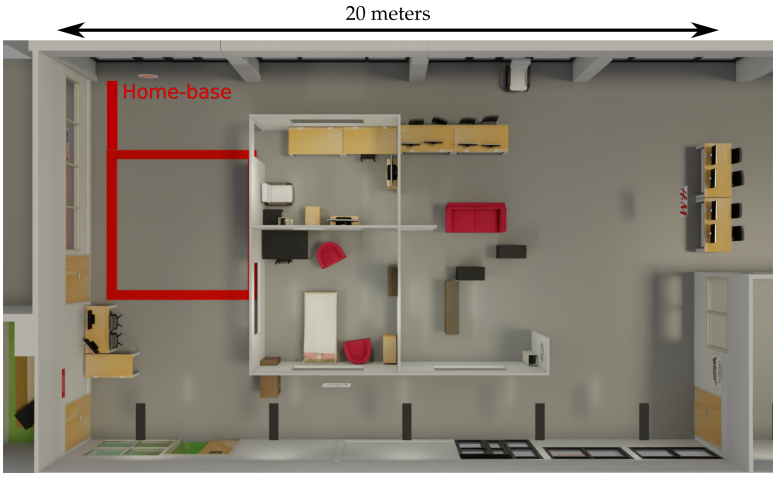
In order for the pointing planner to compute the visibility of the landmarks used for the route description, stairs, escalators, elevators, and store signs are represented each by a single mesh while the rest of the building is a unique 3D mesh. This means that a store is said to be visible if we can see its sign, which we think to be the most relevant element to see to recognize a shop.

The 3D model is also used to generate a navigation map, constraining the robot to move in the interaction area while avoiding obstacles in it.

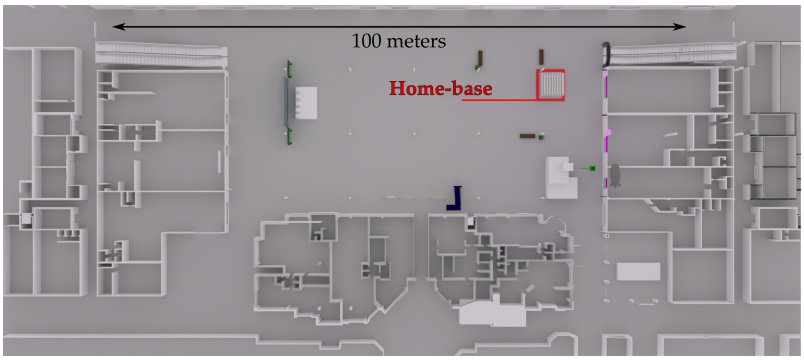
8.5.1.2 Semantic representation

As Satake *et al.* [246], our semantic representation is based on ontology. An ontology allows to define classes representing general concepts (*e.g.*, Restaurant), individuals/entities being classes instantiations (*e.g.*, Burger_King), and properties linking two entities (*e.g.*, Burger_King isIn Ideapark). To provide storage and an efficient way to manipulate the ontology and reason about it, a lightweight software has been developed, called Ontologenius, presented by Sarthou *et al.* [244]. It makes it possible to share the semantic knowledge among all the components of the architecture, here especially the route handler and the supervision, thus enabling a unique repository of knowledge.

The ontology is first used to represent information about the stores. It allows to define and refine the shared goal of the task by understanding the client's wanted



(a) The 3D mesh model of the mockup mall at laboratory. The red square represent the interaction area as a square of 4 meters per 4 meters. Signs representing the shops have been place all around the environment.



(b) The 3D mesh model of the real mall in Finland. The entire mall having a size of 528.6 meters per 247.5 meters on two levels, we have only modelled the part which can be visible from the interaction area. It results in a model of 150 meters per 69 meters.

Figure 8.5: We have built a mockup of the Finnish mall environment in our lab in order to be able to test and debug the direction-giving task in our lab. This environment comprises a two-level area with corridors, "shops" , passages, stairs, open central space and consequently allowed us to run realistic guiding scenarios.

destination. Thus, the stores' types, their names, and the items they sell have been represented in it with a rich semantic. It allows for example to represent that both soda and hamburgers are sold in fast-foods, which are types of restaurants, but that soda can also be found in a supermarket. Thanks to Ontologenius, the names of concepts are defined in different languages and with synonyms for these names. It allows the robot to adapt itself to the human partner language. Moreover, Ontologenius endows the robot with the ability to recognize a set of names in natural language but that it will be prevented to use (*e.g.*, the robot can understand a reference to "bank" when a human says it but only refers to it as "ATM" or "cash

machine" since there was no bank office in the mall). In addition, this software offers a fuzzy match service based on Levenshtein distance, to help the supervision system to handle ambiguities coming from the speech to text component (*e.g.*, it can match the word "Juwelsport" with "Juvesport"). This set of functionalities around the concepts' names facilitates the understanding of the partner's need and thus helps at increasing the quality of interaction.

In an effort to unify representations because representing as a 3D mesh the entire mall would be a complex task, we chose not to use the geometrical representation or a topological map to compute the route to the final goal but rather the semantic representation given by the ontology.

To include topological information into the semantic representation, the Semantic Spatial Representation (SSR) has been designed, presented by Sarthou *et al.* [245]. With the SSR, the overall knowledge is represented in an ontology with three upper classes which are: **region** (*i.e.*, a two-dimensional area that is a subset of the overall environment), **path** (*i.e.*, a one-dimensional element along which it is possible to move and which has a direction) and **place** (*i.e.*, a point of zero dimension that can represent a physical or symbolic element). The **place** class has three subclasses: **path intersection** (*i.e.*, the connection between only two paths and thus a waypoint to go from one path to another), **passage** (*i.e.*, the connection between only two regions and thus a waypoint to move from one region to another like a door, a staircase or a passage), and **shops**. A representation of these classes is visible in Figure 8.6.

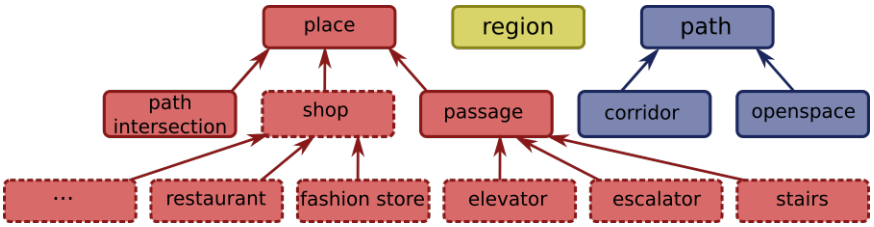


Figure 8.6: Classes for a representation of the topology of an indoor environment in a semantic description. The classes with the solid outline are the minimum classes defined by the SSR. The classes with the dotted outline are an extension of this minimal set.

An example of the final semantic knowledge represented in the ontology for a given shop is presented in Figure 8.7. We find here the identifier of the shop, the category to which each store belongs (*e.g.*, restaurant or hairdresser), the items sold for which people ask the most (*e.g.*, shoes or coat), and the names and synonyms in natural language and that for different languages. Moreover, thanks to the SSR we can produce the best route (in term of complexity) as well as verbalize it using a route perspective.

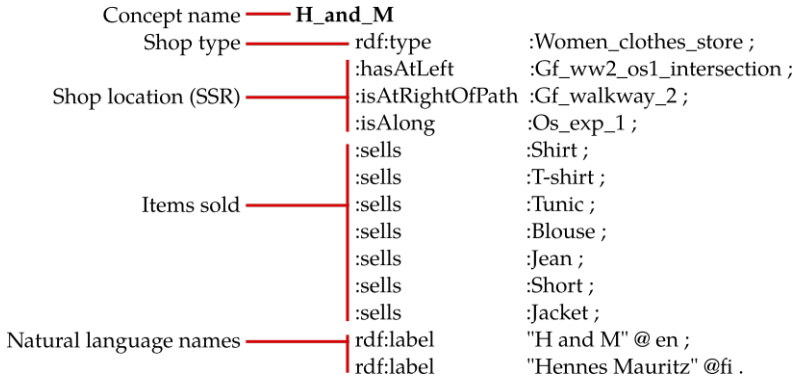


Figure 8.7: Properties for a representation of the topology of an indoor environment in a semantic description.

8.5.2 Perceiving the partner

The situation assessment component is based on the Underworld framework [184]. It aims at gathering perception information in the form of 3D position and orientation of human faces, with the 3D model and the robot state. With this information, it is able to generate the symbolics facts listed in table 8.1.

Predicate	**Description**
isPerceiving	The robot is perceiving a human
isCloseTo	The human is within a distance of 0 to 1 meter of the robot
isLookingAt	The human is looking at the robot
isInArea	The human is in the interaction area
isEngagingWith	The human is close to the robot and is looking at it

Table 8.1: Facts computed and monitored during the direction-giving task.

8.5.3 Managing the robot's resources

A humanoid robot such as Pepper can be seen as a composition of multiple physical components that can act independently of each other. For the direction-giving task, we identified four resources: the head, both arms, and the base. At the beginning of the interaction, for example, the head is used to find people to interact with, but later it will be used to track the human with the gaze. Several components could access this resource to perform these actions. However, they do not have a global picture of the ongoing task. In this case, a resource could be used by several components at the time. Consequently, it could lead to task failures.

Moreover, in some cases, several resources have to be used simultaneously to perform a high-level action. To point to a landmark, one arm is selected to point while the other has to be lowered. The base is then rotated if the arm reaches the joint limit to point a target on its back. If at least one of the involved resources

is simultaneously used to perform another action, the overall high-level action will fail as the global posture will no more be clear. For example, if the human gets too close to the robot and a component tries to move away from a little, the arm would no more point in the right direction.

Thus, for each of the identified resources we instantiated a Resource Manager that we presented in Section 6.5. The global resource management scheme is illustrated in Figure 8.8 with four resource managers and one synchronizer.

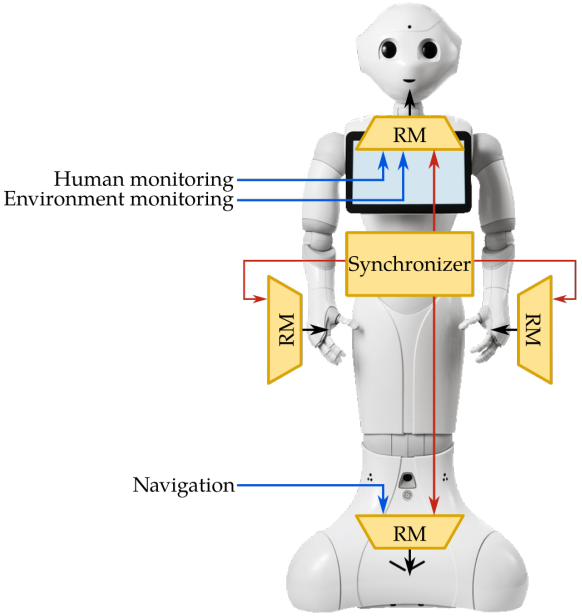


Figure 8.8: Representation of the resource management system with four resource managers and a synchronizer. The red arrows represent the state machines inputs and the blue arrows represent the inputs for permanent commands.

8.5.4 Describing the route to follow

In large-scale environments such as malls, route computation can lead to combinatorial explosion. Therefore, to simplify the problem we chose to divide it in two stages and to conceive an algorithm for each one. The first one computes the existing routes from one Region of the mall to another. A region is for example a floor of the building so if the robot is at the ground floor and the final destination also, this means that the algorithm will not take into account all the elements of the other floors. Then, the second algorithm uses the Region-to-Region routes to calculate the Place-to-Place route. These algorithms are presented with more details in [245].

Region-to-Region route In the SSR, **passages** (*e.g.*, escalators, stairs) are elements of the environment connecting two regions through the $isIn$ property. With this property and a breadth-first search algorithm, Region-to-Region route finding algorithm is able to find the routes connecting two regions using only passages. It

outputs a route with the format $region - place - region - ... - region$. In the example of Figure 8.9, the final routes found by the algorithm to go from Region 1 to Region 3 are:

- $region\_1 - passage\_1 - region\_2 - passage\_2 - region\_3$
- $region\_1 - passage\_1 - region\_2 - passage\_3 - region\_3$

Place-to-Place route The Place-to-Place route search is based on the Region-level search results. It decomposes each route into sub-routes of the form $place - region - place$. In our example, the division gives five unique sub-routes:

- $start - region1 - passage\_1$
- $passage\_1 - region\_2 - passage\_2$
- $passage\_2 - region\_3 - end$
- $passage\_1 - region\_2 - passage\_3$
- $passage\_3 - region\_3 - end$

Then, the algorithm aims to replace each sub-route region with a succession of paths and intersections. It works on the same principle as the previous search algorithm using the $isAlong$ property instead of the $isIn$ property. Still taking the same example and focusing on $region\_1$, it can solve the sub-route $start - region\_1 - passage\_1$. $Region\_1$ is represented with its corridors and intersections in Figure 8.10. By applying the breadth-first search algorithm at the Place level, a solution of the form $place - path - place - ... - place$ is obtained. So for our example, $start - corridor\_1 - intersection\_1 - corridor\_5 - passage\_1$ is a solution for the first sub-route. By doing the same for each sub-route, we can then recompose the global routes and give a detailed set of routes from start to end.

The second place of the route – the third element of the route objects – is the one we call the passage in the description of the direction-giving task, the first salient landmark of the route to point to, which is on the way to reach the final place, *i.e.*, $intersection\_1$ in the example.

8.5.5 Planning a shared visual perspective

When the robot has to point to a target, two criteria have to be respected. First, the human has to be able to see the target. Second, the human has to be able to

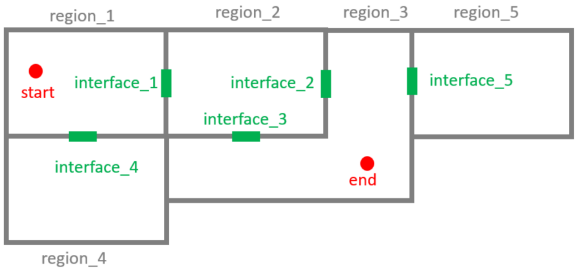


Figure 8.9: Representation of an environment at the regional level.

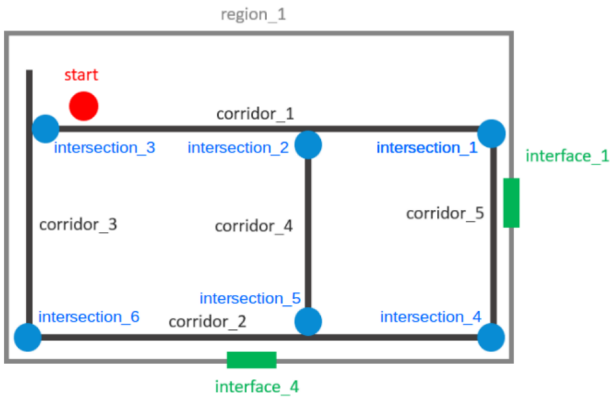


Figure 8.10: Representation of corridors and intersections in region_1

look at the pointed target and at the robot without turning the head too much. It goes the same for the robot as it has to see the pointed target, meaning not to point toward a wall and be able to simultaneously point at the target and look at the human. Consequently, to point a target in its back, it has to move. The robot and the human can thus move in the interaction area during the direction-giving task, to move to a better position for pointing at the target. To find the robot and human possible positions we designed a component called the SVP (Shared Visual Perspective) Planner, presented in [299]. For the purpose of the deployment, the presented version is an adapted and slightly simplified version.

To compute the visibility of both agents, the planner has access to the geometrical representation of the environment and the agents current positions. In addition, it considers an estimated agent's maximal speed to move and a visibility threshold.

When the robot explains the route to the human and points to a landmark, they form what is called an F-formation. Kendon explains that *"An F-formation arises whenever two or more people sustain a spatial and orientational relationship in which the space between them is one to which they have equal, direct and exclusive access"* [153]. This F-formation has been decomposed by McNeill into two types: the social formation and the instrumental formation [196]. While the first type corresponds to the original definition, the instrumental formation includes a physical object that all the agents can gaze at. This means that once the robot will have moved, the human will come in front of it creating a social formation in the form of a vis-a-vis (each facing the other) and when the robot will point they will change for an instrumental formation. Indeed, when both agents will reach their position computed by the planner, we want them to be able to go from one formation to the other with only a rotation; the human will not need to move again from their arriving position to see what the robot will point.

To search for better positions to reach in order to point a landmark, the planner takes three main parameters into account:

- Visibility constraint: The two agents can see either the target shop when it

is the only element of the route or the passage.

- Navigation distance cost: The agents do not have to move too much.
- F-formation cost: The human-robot-target angle and a robot-human-target have to be less than 90°.

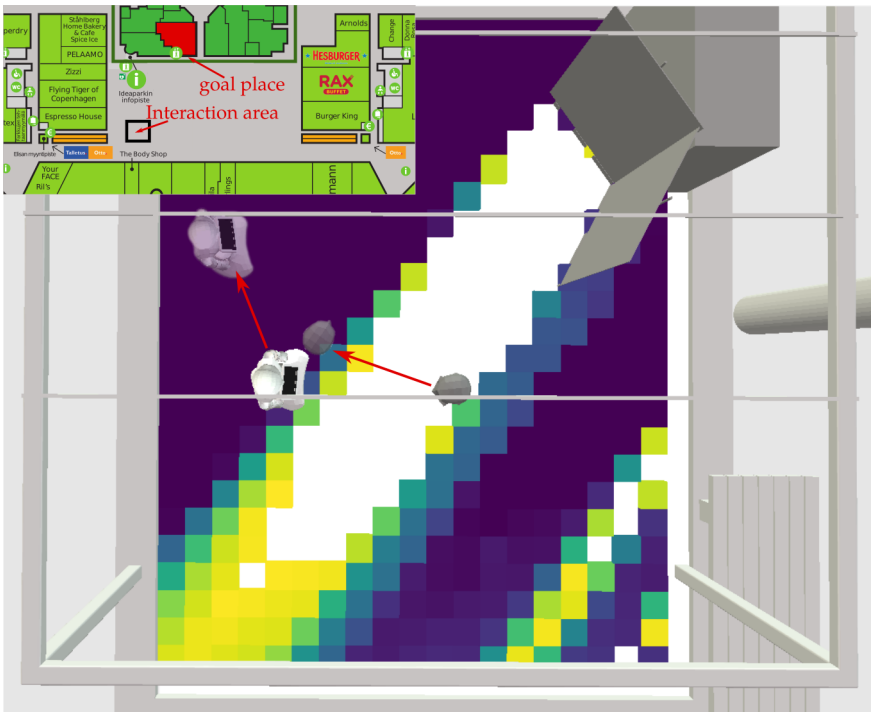


Figure 8.11: Visibility grid for a target located at the top right. The uncoloured areas represent an absence of visibility and the others represent the cost of visibility ranging from yellow for low visibility to purple for good visibility. The robot and the human in transparency on the image represent the final calculated positions while the others are the initial positions.

To compute the positions, the interaction area is firstly decomposed into a weighted three-dimensional (x,y for the possible positions in the area and z for the human height) grid representing the estimated human visibility of the target. The target visibility is computed offline for each position of the grid. It is based on the part that the target takes in the 360° field of view of the environment. Such grid is represented in figure 8.11 for a given human height. The white cells are positions from which the human cannot see the pointed target. The other colored cells represent the degree of visibility from the poor in yellow to the good in purple. Having the human visibility grid, the goal position is computed using a weighted cost function between good visibility and restricted distance to cross. In the example of figure 8.11, the transparent human head is the human goal position while the other is the initial position. From the initial position, the human was not able to the pointed target.

The robot position is computed in a second time, according to the human planned position. Divided the search into two steps allows reducing the search complexity. The robot position is thus constrained by the human one. It has also to respect a minimal and maximal distance to the human and minimal visibility of the target from it. Finally, the robot position is also determined regarding a cost preferring an F-formation limiting the robot reorientation, meaning that it can point to the target keeping its torso and its chest oriented towards the human.

8.5.6 Navigate close to human

The Human-Aware Navigation component aims at moving the robot while avoiding dynamic and static obstacles in addition to proposing a socially acceptable navigation solution for the robot. For example, the robot should not pass too close to the human and should not show its back while navigating around the human. A full presentation of the planner is available in [270].

8.5.7 Robot execution control and supervision in a joint action context

The work presented about the supervision in this section is an early version of Joint Action-based Human-aware supeRVISor (JAHRVIS) presented in Chapter 6. It integrates on one hand the decision and control for the direction-giving task, and on the other hand the implementation in this task context of the metrics presented in Chapter 7, to measure the Quality of Interaction.

8.5.7.1 A supervision and control system dedicated to human-robot joint tasks

A service robot interacting with humans in a mall and providing directions to them needs a number of abilities to enable a smooth and efficient interaction. As explained in Section 8.3, the direction giving task is an asymmetric joint action, with the robot in the guide role and the human in the guided role. The *Supervisor*, the supervision and control system of the robot, is built taking this specificity into account, embedding a shared representation of the direction giving task. More specifically, when giving directions to a human, the robot plans its actions and the human ones and then execute its part of the plan. To be able to know if and when the human performs their actions, it monitors the action executions and interpret the information received from the Situation Assessment (see Section 8.5.2). Furthermore, in such interaction, communication is important, thus the robot communicates verbally as well as non-verbally, and listens to the human. All along the interaction, it needs to maintain a distinct mental state model for the human and itself concerning the knowledge of both agents and the state of the world. Finally, it should be able to tackle events and contingencies happening during the task and to drop it when necessary.

During an interaction session (see Section 5.2), *direction-giving task* occurs when the human involved in the ongoing interaction session asks for directions to a place or for locations of sold items.

8.5.7.2 Implementation of the direction-giving task and its associated actions

In the direction-giving task, plans were not computed by a planner as presented but were written with Jason reactive plans (see Section 4.3.2). Thus, at execution time, the Supervisor does not handle one shared plan received from a planner but plenty of (reactive) plans which are chosen among the ones from the plan library when triggered by an event or by another plan. The same plan can have multiple versions and the version to be executed is selected according to the pre-conditions (also called context). For instance, the plan *verbalization(Target)* has two different versions, one in the case where the target to point is visible and the other one in the case where it is not, and at execution time, the selected one will depend on the presence or not of the belief `visible_target(Target)` in the Supervisor belief base, as shown in Listing 8.1:

```
+!verbalization(Target)                    // plan name
: visible_target(Target)                 // context
<-  ?verba_name(Target, Name);           // belief query
say(visible_target(Name)).           // action

+!verbalization(Target)
: not visible_target(Target)
<-  ?verba_name(Target, Name);
say(not_visible_target(Name)).
```

Listing 8.1: Two different plans for `verbalization(Target)`

Even though the direction-giving task is implemented with reactive plans, it can still be represented with an activity diagram, for presentation purposes. This activity diagram is visible in Figure 8.12. Each frame represents one of the steps described in Section 8.4.2. We now present their internal functioning and the interactions with the multiple components of the system the Supervisor has.

Establishing the shared goal When a person triggers a direction-giving task, they might directly ask something like "where is the pharmacy?" which allows the robot to directly establish the shared goal but, they might also ask something less precise. In the latter case, the robot needs to inquire about the human desired place to reach in order to establish the shared goal.

When a person asks "Where is a good restaurant?", the robot presents a list of the types of food available, namely "There are casual dining restaurants, Asian restaurants, native food restaurants, hamburger restaurants, fast food restaurants,

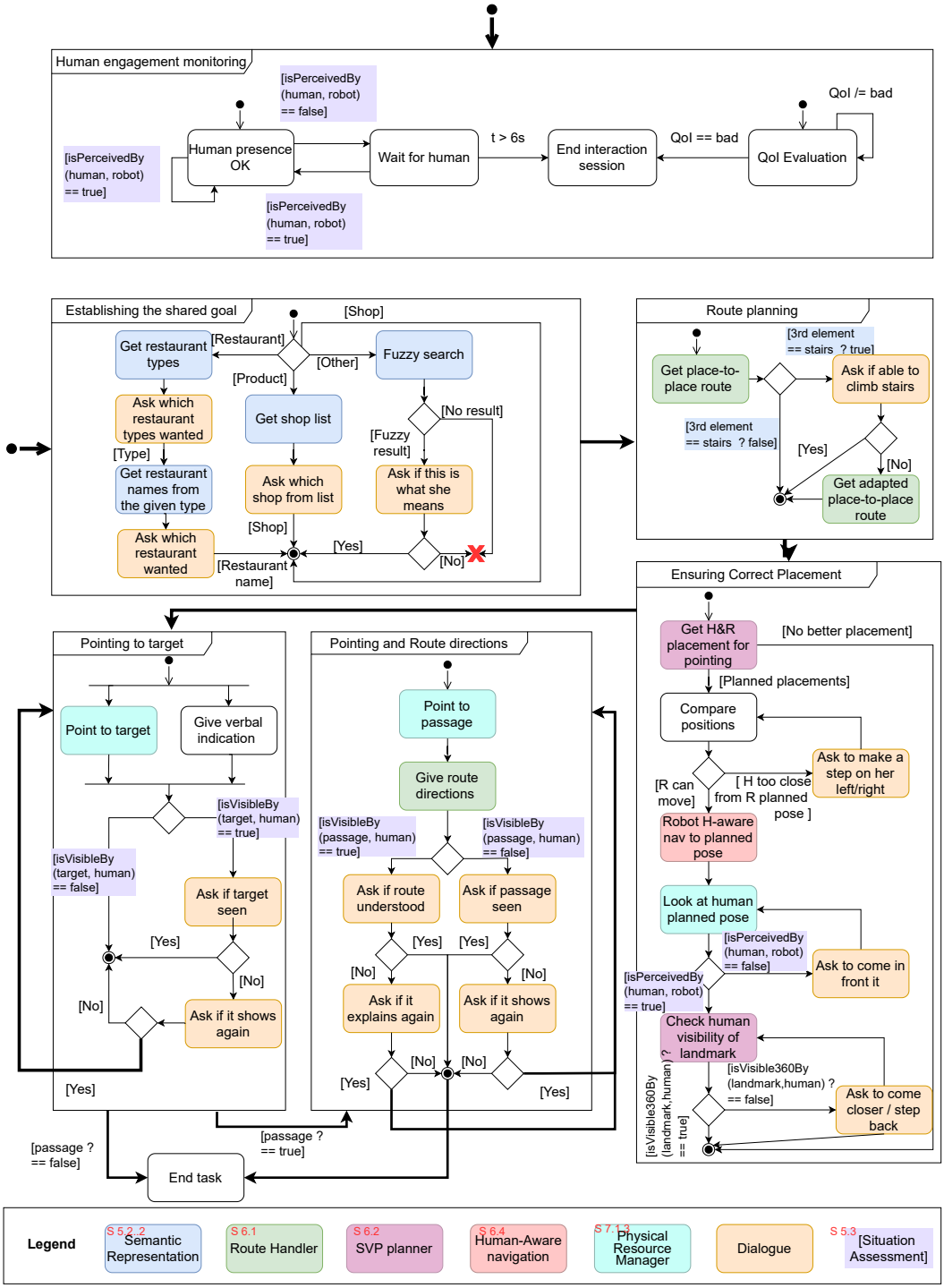


Figure 8.12: Supervisor activity diagram of the direction-giving task. Each action has a color corresponding to the component with which the Supervisor interacts to execute it. It goes through every subtasks described in Section 8.5.7.2. Also, the human engagement monitoring is represented. Texts between brackets correspond to beliefs on which depends the decision-making process. These beliefs can either be provided by other components or being the result of the Supervisor's own computations.

and pizzerias.". This behavior is quite similar to the recommendation behaviors of Kanda *et al.* [148].

To be able to display this behavior, several components of the system are requested. When the Supervisor receives $\{request = restaurant\}$ as data from the Dialogue, it asks Ontologenius (see Section 8.5.1.2) for all the existing restaurant types. This list of restaurant types is sent to the Dialogue whose role is to return to the Supervisor with the type selected by the human. Finally, similarly to the way it obtained the restaurant type from the human, the Supervisor tries to get the restaurant name. Therefore, it requests from Ontologenius all the restaurants serving the given type of food. Then, this list is sent to the Dialogue whose role is to return to the Supervisor with the restaurant selected by the human among the elements' list. It should be noted that all the restaurants of the given type are suggested to the person, even though sometimes the list is long. We thought of alternatives such as randomly giving three restaurants among the ones of the list. However, these alternatives were not allowed by the mall policy as they could not provide equality between all shops.

The same principle goes for products. For example, people can ask "Where can I buy a dress?". Then, the Supervisor gets from Ontologenius a list of shops selling dresses and passes it to the Dialogue. The Dialogue returns the name of the shop chosen by the person.

When the Supervisor receives as a goal a name it does not understand, it queries Ontologenius to try to match it to a known name as it may be not understood because of a speech recognition failure or a shortened name. For instance, thanks to the fuzzy match provided by Ontologenius, when a person asks to go to "jewelsport", the system can make the assumption that the person actually asked for "Juvesport". So the robot asks the person, "do you mean Juvesport?", to which the person can answer "yes" or "no". If yes, it starts the direction-giving task, if no it drops it and returns in chat mode.

Enquiry about human willingness and abilities to climb stairs As the robot is there to help humans, it has to adapt to their abilities and preferences such as a person with a shopping trolley will prefer to take escalators than stairs. The preferences definition is currently done through verbal communication.

To determine human preferences about stairs, the Supervisor first requests to the Route Handler (see Section 8.5.4) the possible routes to go to the target shop. The returned routes are of the form $place-path-place-...-place$. The Supervisor selects the one with the smallest cost and then checks if one of the *place* elements is stairs (*i.e.*, the Supervisor queries Ontologenius for the element type). If it is the case, the Supervisor asks the Dialogue with finding out if the human is able to climb stairs or not. If not, it will send a new request to the Route Handler with the parameter "no stairs" and will get a new set of routes. The Supervisor selects the one with the smallest cost. This new route will have a cost equal to or higher than the first one (since it was not the route with the smallest cost in the initial request),

which means the goal might be more complicated to reach or it might take more time.

Ensuring a correct placement The robot's role in this task is not only to give verbal route directions but also to point to the target and the passage (*i.e.*, the third element of the route as explained Section 8.5.4) the person should take in order to increase the chances that they reach their destination as it helps to orientate them in space. For the pointing to be as efficient as possible, the robot computes new positions for itself and the human where the visibility of the pointed landmarks will be better (when feasible). Then its goal is to have itself and the human reaching these new positions.

In the first step of this subtask, the Supervisor requests from the Shared Visual Perspective (SVP) Planner (see Section 8.5.5) the new positions for the robot and the human, with the passage to point (or the target if no passage) and the human identifier as parameters. Then, the Supervisor compares the newly received positions with the current ones of the human and the robot – the current position of the human is provided by the Situation Assessment. In the case where the robot planned position is very close to the human's current position (< 0.5 m), the robot asks the human to step aside on the right or left, depending on the human's planned position. If the human does not move or does not go far enough from the planned robot position, the robot will ask again.

Then, the Supervisor requests the Human-Aware Navigation (see Section 8.5.6) to move the robot to its planned position. Once the Human-Aware Navigation returned that the position has been reached, the Supervisor looks for the human. It is a form of monitoring, which we show in Section 8.3 is important in a joint action. If the human is not perceived – the Supervisor did not receive from the Situation Assessment the predicate isPerceiving$(robot, human_i)$ – in the following seconds (6 seconds in the deployed version), the robot asks the human to come in front of it – this is the way we have chosen after several trials (other modalities like indicating to the human by a gesture where they should stand were not sufficiently successful). If the human is still not perceived after a few seconds, the robot will ask again, remaining engaged in their joint action for a while before giving up.

Once the human arrives in the robot field of view – which means that the human more or less reached their planned position since the robot is looking in the direction of it –, they might not exactly be at their planned position. In this case, their position may not be suited to properly see what the robot has to point at. To check if they are in a position good enough to see, the Supervisor asks the SVP Planner for the visibility (at 360 degrees) of the landmark to point. In the case where the SVP Planner returns that the landmark is visible, the interaction continues. Else, the robot asks the human to move forward or backward in order to adjust their placement according to their planned position. This stops when the robot computes that the position of the human will allow them to see the target. In this way, the robot tries to ensure to put the human in the best conditions as

possible for the next steps, using key elements of the joint action: monitoring of the partner actions', sharing a visual perspective and showing engagement in the task.

Pointing to target As it is shown that the use of deictic gestures such as pointing improves the understanding of route directions (see Section 8.4.1), we endowed the robot with this ability.

To do so, the Supervisor requests from the Physical Resource Manager that the robot points to the target. At the same time, it generates a short sentence for the robot to say and sends it to the Dialogue. The sentence varies according to the visibility of the target such as "Here, you can see Burger King" for a visible place and "The restroom is in this direction" for a non-visible one. In this way, the robot shares the human's perspective and takes into account the knowledge they can get from their environment in respect of the joint action principles. In this way, the human knows if they have to try to notice it from their place or take this information as an orientation indication. In order to continuously look at the human and not loose them from its sight, the robot does not turn its head towards the target when pointing.

It is important for the robot to know if it successfully communicated the information to the human. Then, it asks if the target has been seen, as it wants to ensure its action had the expected effect.

Pointing to passage and giving route directions This step is executed when there is a passage in the route returned by the Route Handler. Therefore, the Supervisor sends a route to the Route Handler which returns a verbalization of this route (*e.g.*, "Walk through that corridor, and then, turn left. From there on, Apteekki will be on your right, straight after Glitter"). Then, as explained in the *Pointing to target* paragraph, the robot points, to the passage this time. And, at the same time, it verbalizes the route received from the Route Handler, added "in this direction" to the sentence if the passage is not visible.

As for ensuring the target has been seen, the robot wants to make sure it has been understood and leaves the possibility to the human to hear the route directions again if they need it. In the early versions, we had programmed the robot to ask if the passage had been seen and then if the route had been understood but it was too many questions that seemed useless to users. Indeed, we analyzed it as a postcompletion error [57], as the goal of the human was to know the route to their location, whatever actions arising after this goal has been completed are often forgotten. In the end, the first question is asked in case of a visible passage and the second one is asked in case of a non-visible one.

It may be noted in Figure 8.12 that it is possible to go in infinite loops such as Route directions - Ensuring route understood - Route directions - ... . To avoid this issue, the Supervisor prevents to return inside a step if it has already been executed a certain number of times (in the final version, 3 was the maximal number a step could be executed).

8.6 The deliberative architecture in a real-world environment

In the previous section, we presented a deliberative architecture designed to be embedded in a service robot. The purpose of this robot was to be deployed in a mall in Finland. To make this deployment successful, we did extensive tests in our laboratory where we had reproduced a part of the mall environment to be in the most realistic conditions possible[6]. Some of these emulated shop are visible in Fig. 8.13. In Sect. 8.6.1, we introduce the environment setup as well as the robot one. Then, in Sect. 8.6.2 and Sect. 8.6.3, we present our tests and deployment in the Finnish mall.

8.6.1 Environment and robot setup in the Finnish mall

Our architecture has been tested and deployed in a mall in Finland. As we explained previously, it has two abilities: chat with people and guide them, but in this paper we consider only the latter. The robot was able to interact in English and Finnish, though due to the vast linguistic differences between the two languages, the two versions have been kept separated, and the whole interaction can either be in one or the other.

8.6.1.1 The robot home-base

For availability for as many customers as possible, the robot was contained in a defined place in the mall as shown in figure 8.14. A home base was designed with the participation of all the project partners. It was a 4 per 4 meters area with a 2.5m high frame structure on it. The home base included a non-reflecting carpet on the floor and an acoustic ceiling surface on the roof.

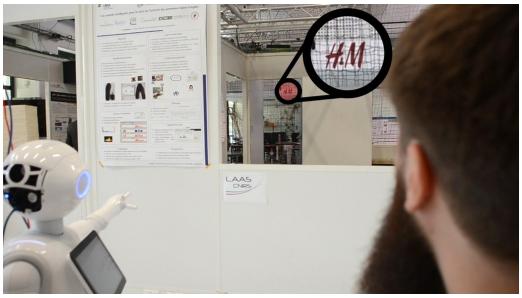
During the first deployment in the real mall, we have updated both the Geometric Representation with actual measurements and the Semantic Spatial Representation (SSR) by making sure the regions, interfaces, corridors and intersections were represented reflecting the actual mall topology. To ensure the correctness of the instructions given by the route handler, we generated routes from the deployment location to several shops in the mall and followed them to the destination. Inaccuracies, as well as algorithmic flaws, have been fixed using this method. We also tested the interaction in the Finnish language with our native Finnish partners and corrected some mistakes in the route verbalization.

8.6.1.2 Hardware architecture

The robot is an upgraded, custom version of the Pepper platform [59], which is equipped with an Intel D435 camera and an NVIDIA Jetson TX2 in addition to the traditional sensors that are found on the previous versions of the robot. We

[6]This setup not only was used for tests but also for public demos and even in the context of a scientific live event now accessible on `https://youtu.be/p4f3iwHht2Q?t=4495`

(a) A person being guided, the emulated shop "Zizzi" is visible in the background of the picture.



(b) A person being guided, the emulated shop "H&M" is visible.

(c) Two people simulated going to shop. The emulated shop "Burger King" is visible in the background and a small part of "Thai Papaya" is visible in the foreground.

(d) A person being guided, a sign towards the toilet and the shop "Marco Polo" are visible on the left of the picture.

Figure 8.13: Examples of emulated shops of the Finnish mall in our lab.

Figure 8.14: The pepper robot in its interaction area in the Finalnd mall, Ideapark.

used the Robot Operating System (ROS) to enable inter-process communication between the processing nodes. All the streams (audio, video, robot states) are sent to a remote laptop which performs all the computation. The laptop has an NVIDIA RTX 2080 graphics card (for the visual perception system) and 12 CPU cores. The 4 microphone streams are processed at a frequency of 16000 Hz, and the full perception system delivers the output at 10 fps.

8.6.2 Pre-deployment in the Finnish mall, in-situ tests

Three integration sessions, each lasting one week have been made on site, in September 2018, June 2019 and September 2019, in the mall in Finland. The whole LAAS developer team were part of these integration weeks, along with our project partners. So, I spent around 150 hours (3 times 5 days) in the mall for software integration debugging with the other developers, and testing and debugging of the direction-giving task. During the integration weeks, only expert users (developers) interacted with the robot for testing purpose.

To have a working system in the lab and to have a working system in a real-world site are two different things. As much as a team prepare for an in-situ deployment, there will always be elements that will need to be tuned on site and unexpected bugs arising. Thus, I had to handle a lot of contingencies, diagnosing where the issue came from, repairing if it was originating from my software, communicating with the person responsible for the component having a bug if it was not from mine, and testing again.

The first step to perform for us once on site was to update both the Geometric Representation (see Sect. 8.5.1) which was previously based on architectural plans and refined with actual measurements and the Semantic Spatial Representation (see Sect. 8.5.1) by making sure the regions, interfaces, corridors and intersections were represented reflecting the actual mall topology.

Finally, to ensure the correctness of the instructions given by the route handler,

we generated routes from the deployment location to random shops in the mall, and followed them to the destination. SSR inaccuracies as well as algorithmic flaws have been fixed using this method. We also tested the interaction in Finnish language with our native Finnish partners and corrected some mistakes in the route verbalization.

8.6.2.1 Component integration problematic

Even though components were integrated together before getting on site, code modifications as mentioned above and intense testing can make new issues appear. So, it was essential to test the integration between all the components after this.

Finally, once everything was running quite nicely, some time has been dedicated to fine-tune the direction-giving task, ensuring all the components could withstand running for several hours in a row, with naive users possibly interrupting the task at any stage.

8.6.3 "In the wild" deployment

The robot was then installed for a long-term 14 weeks deployment from September 2019 to December 2019. During this period, the robot interacted with everyday clients of the mall, who may never had the chance to interact with a robot before. The robot was active for 3 hours per day, three days a week. As it was a project with multiple partners, it was not always possible to have our direction-giving task running. The direction giving task has been available on the robot 32 days out of the 42. The days during which it was not running, the partners' software executing on the robot where the visual perception and the dialogue that we mentioned previously, and a social signal processing feature [98].

Nowadays, having an autonomous robot in the wild is a challenge. At first glance, we could think that if the robot is able to run smoothly for a few hours, the challenge would be met. However, there are a lot of other elements to take into account. First, how to guarantee the safety of children and elderly? How to ensure that the robot will not fell on or bump into them despite the robot sensors, hurting them? Furthermore, not only people safety is important but making sure that the robot is not damaged by people as well. People might indeed be brutal towards the robot, on purpose or not.

To tackle the "obvious" issue, making sure that the robot continuously running, it was remotely watched by an on-call developer of the project team. At the beginning of the time slot, they launched all the software on the robot. Then, they checked through component monitoring, time to time, if everything was running properly, and they were in contact with the robot guard who told them if she noticed something wrong with the robot. They also had access to a video feed of the robot home-base if needed. Thus, all along this long-term deployment, we adjusted parameters and fixed bugs, with the help of the on-site team VTT and the robot guard that tested the direction-giving task when we asked her. The bugs we en-

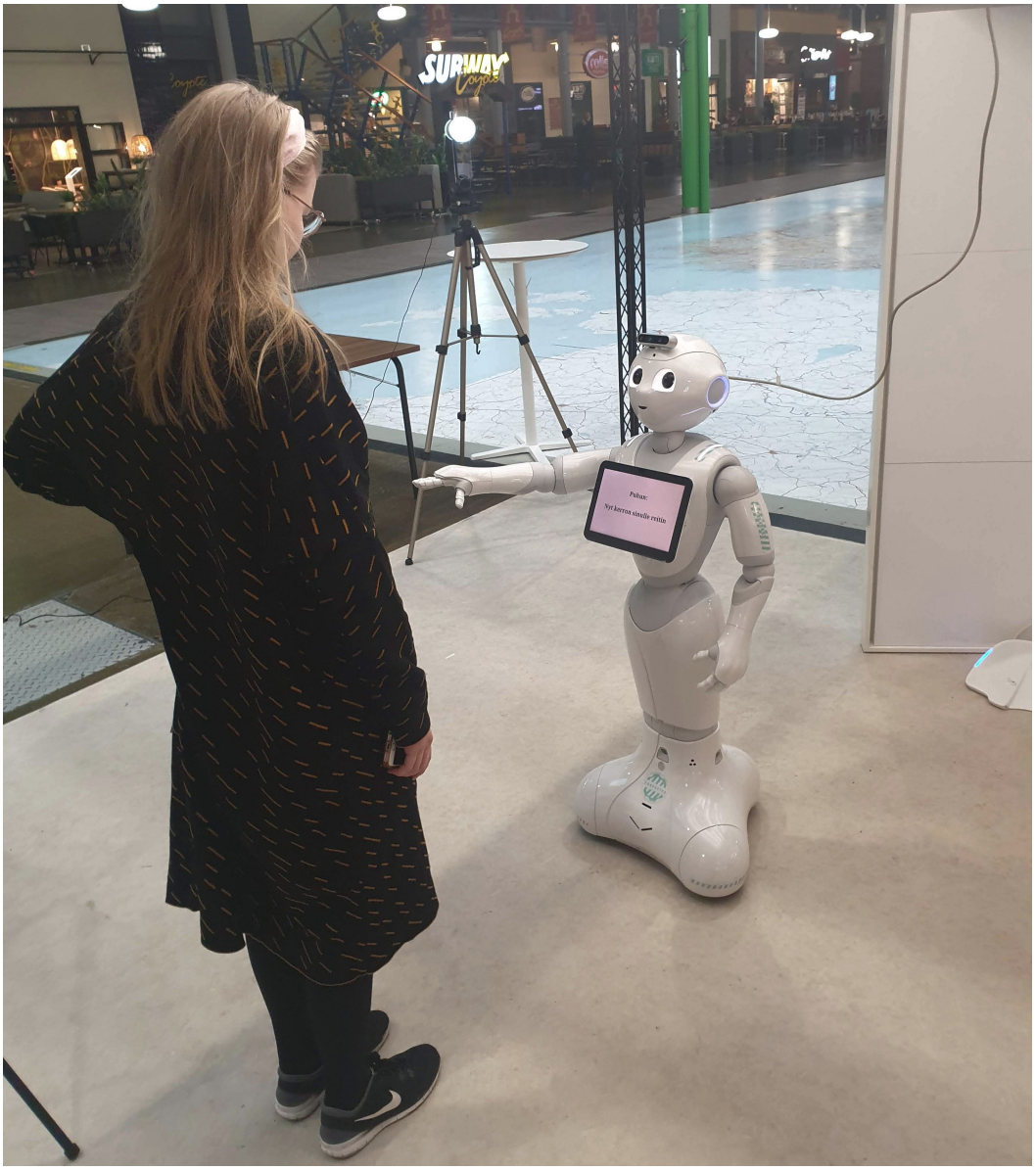


Figure 8.15: A person receiving directions from Pepper. (Image from VTT team)

countered concerned mainly Finnish translation issues (*e.g.*, "just after Arnold's" was translated "paikan päälle Arnolds" in Finnish but the correct way to say it in Finnish was "paikan Arnolds jälkeen" thus we changed the English sentence into "right after the place Arnolds" to be able to get this translation), shop names issues (*e.g.*, Finnish people use the utterance "Hennes Mauritz" and not "H&M" which was the name in the robot ontology originally) and route issues (*e.g.*, a route has one more turn than it should have).

The project consortium tackled the two safety issues (people safety and robot safety) by hiring a "robot guard" and by putting a sign notifying parents to not leave their children alone with the robot. During the robot active hours, this guard employee was physically present to ensure people were respectful towards the robot, *i.e.*, not hitting it or pulling it, to watch the kids who may get too close to the robot when it could have moved so they would not risk to be hurt, and to answer people who wanted to know more about the robot or the project than what was explained on the explanatory posters. She was also responsible for starting and shutting down the robot at the beginning and at the end of the half-day. Besides, for security and legal responsibility reasons, we chose to not have the robot navigating during this deployment as it would have been a complicated issue if the robot bumped into someone, especially a kid who could be hurt. It would have been possible if Pepper had a remote emergency stop which could have been given to the guard. Therefore, the step *Ensuring correct placement* was removed in this context. Then, the Human-Aware Navigation component (see Sect. 8.5.6) was disabled and the Shared Visual Perspective Planner (see Sect. 8.5.5) was only used to compute the 360 degrees visibility of a landmark from the person position.

In total, the robot ran the direction-giving task during approximately 96 hours "in the wild". Out of these 96 hours, it was interacting with someone during 45 hours. Table 8.2 summarizes statistical data about the interaction sessions and Table 8.3 summarizes statistical data about the direction-giving tasks.

Description	Value
Number of occurred interaction sessions between a human and the robot	979
Cumulative duration of the interaction sessions	2720 min
Minimal duration of an interaction session	0.1 min
Maximal duration of an interaction session	41 min
Average duration of an interaction session	2.8 min
Standard deviation of sessions duration	3.3 min
Average number of direction-giving tasks during a session	1.1
Percentage of sessions terminated by goodbyes	30%
Percentage of sessions terminated by the participant not perceived by the robot anymore	70%

Table 8.2: Statistics on interaction sessions in the wild

Description	Value
Number of occurred direction-giving tasks between a human and the robot	1156
Cumulative duration of the direction-giving tasks	930 min
Minimal duration of a direction-giving task	0.01 min
Maximal duration of a direction-giving task	22 min
Average duration of a direction-giving task	0.8 min
Standard deviation of direction-giving tasks duration	1.27 min
Success rate of the step *Establishing the shared goal*	63%
Success rate of the step *Route planning according to the human willingness and ability to climb stairs*	100%
Success rate of the step *Pointing to target*	56%
Success rate of the step *Ensuring target seen*	39%
Success rate of the step *Pointing to passage and giving route directions*	94 %
Success rate of the step *Ensuring passage seen or route understood*	92%
Success rate of the removed step *Check if indications understood*	19%

Table 8.3: Statistics on the direction-giving task in the wild. *Ensuring target seen* is a part of the step *Pointing to target* as described in Sect. 8.4.2. Likewise, *Ensuring passage seen or route understood* is a part of the step *Pointing to passage and giving route directions*. The success rate of a step is the number of times the given step has been achieved over the number of times it was planned (*e.g.*, *Route directions and pointing* is not planned if there is no passage to point), all direction-giving tasks combined. Steps were not achieved sometimes because of robot failures but most of the time it was because the human was leaving during the task. As mentioned in Section 8.5.7.2, we did not keep the step *Check if indications understood* all along the deployment because, as shown by the success rate, people were leaving before answering this question. Then, as this step was considered as superfluous by users, we merged it with the one before, *Ensuring passage seen*.

8.7 Integration and test of the QoI Evaluator

As a proof-of-concept for the QoI Evaluator presented in Chapter 7, we integrated it in the direction-giving task described in this chapter. It is also an excerpt of the paper accepted in the Journal of Social Robotics [195].

More specifically, this implementation of the Quality of Interaction Evaluator measured the interaction quality at the direction-giving task level and at the elementary actions level, omitting the interaction session level as this latter was not our focus in the MuMMER project. The QoI Evaluator was integrated into JAHRVIS presented in Chapter 5. The QoI Evaluator is implemented into a Jason function (the reasoning cycle) which is invoked periodically. After multiple testings, we reached the conclusion that it was pertinent, at least in the context of the direction-giving task, to have the Evaluator computing the QoI every second for both levels. Therefore, every second, the system computes the value of each metric and then outputs a value for QoI_{task} and QoI_{action}.

As mentioned in the step 4b of the chronicle, the robot interacted in the wild with dozens of usual customers (Fig. 8.16), executing around 350 direction-giving tasks. This allowed us to improve the performance of the direction-giving task, to gather standard durations of the subtasks executions and to draw lessons about metric definitions and choices (*e.g.*, we realized it was not relevant to measure the human visual attention towards the robot when it was giving the route explanation as humans look around at this moment). Unfortunately, the practical conditions of the project deployments did not offer us the possibility to evaluate the QoI Evaluator based on a study in the mall with real customers. So, we demonstrated – after improvements of the metrics equations such as the Distance-to-Goal one, and manual tuning of their parameters based on the experience in the mall – our finalized concept through tests in our lab (step 6). This is shown in Sect. 8.7.3 where we present and discuss, a comparison of the QoI computed by the robot when it is dealing with an "ideal" human, a "confused" human and a "non-compliant" human during a direction-giving task, performed in the lab. Before that, we present in Sect. 8.7.1 and Sect. 8.7.2 how the QoI is evaluated at both task and action levels for the direction-giving task.

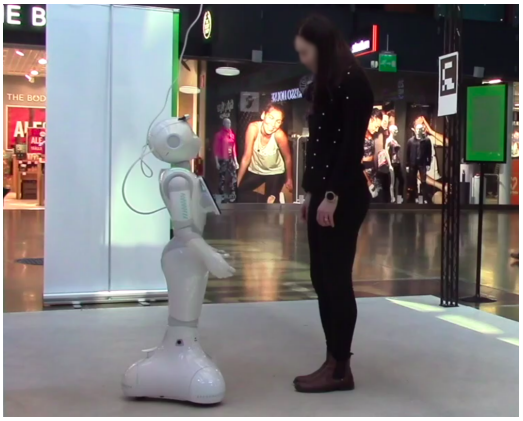
8.7.1 QoI Evaluation at the task level

In the context of the direction-giving task, we have selected two metrics to evaluate the QoI at the task level: a metric defined in the Sect. 7.4, the *Deviation from standard duration* and, the aggregation over time of the actions QoIs. Following the process of Fig. 7.1, we measure the QoI of the Task_i = direction-giving_task, based on the QoI of all task actions and Task metric_1 = Deviation from standard duration.

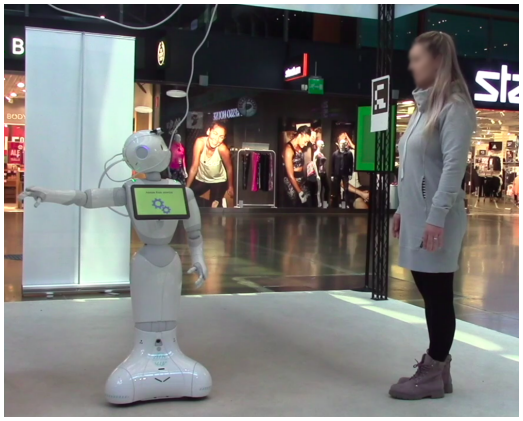
The *Deviation from standard duration* is used to measure the QoI at the task level as the task is a sequence of subtasks. Indeed, if the subtask lasts longer than expected, the QoI should decrease. Then, as needed for the metric computation we

(a) A customer listening to Pepper after repositioning

(b) A customer listening and Pepper pointing to a corridor



(c) A customer answering to Pepper



(d) A customer listening and Pepper pointing to a shop

Figure 8.16: MuMMER robot engaged in direction-giving tasks. Around 350 trials with customers in the mall allowed us to gather empirical data to select the metrics and tune the measuring functions parameters.

have determined the values of the soft deadlines SD_i for each subtask $a_i, i \in [0, 4]$, using the empirical data we gathered as explained in. Specifically, we have computed the average time execution of each subtask, after removing the cases for which the execution of the subtask was annotated as not smooth. These soft deadlines are presented in table 8.4. Finally, we chose $V_i = 0.5$ for all the subtasks.

ref sec

Subtasks	soft deadline (s)
Target refinement process	30
Ensuring Correct HR Placement	30
Ensuring target seen	20
Direction explanation and pointing	30
Ensuring Direction Seen	20

Table 8.4: Soft deadlines SD_i for each subtask of the direction-giving task

The task QoI is also dependent on the actions QoI values (their computation is described in Sect. 8.7.2). Indeed, the actions QoIs should be reflected on the task QoI as, if a majority of the actions have a low QoI, the task QoI cannot remain high. That is why, besides the *Deviation from standard duration*, we take into account the average of the action QoIs of the actions already executed or still running.

Then, the task QoI is computed using Equation (7.1) presented in Sect. 7.3. After various trials we have empirically chosen the weights W_i for each metric $M_i, i \in [0, 1]$. The final equation to compute the task QoI is:

$$QoI_{dir-giv\_task}(t) = \frac{\Phi_{dir-giv\_task}(t) + 3 * \overline{QoI}_{actions}}{4}$$

8.7.2 QoI Evaluation at the action level

As mentioned earlier, each subtask of the direction-giving task can be decomposed into actions. These actions involve several turn-taking steps, the robot asking complementary information, informing the human or expecting an action or reaction from them. We need to measure the QoI during the execution of each action. To do so, we have chosen one or more metrics for each action.

For each action of the following list, we explain which metrics M of Table 8.5 we have used and scaling functions of Appendix B and then, how we compute the action QoI. Finally, the ways metrics are aggregated for each action, outputting QoI values, are summarized in Table 8.7.

(a) *Robot-Human information sharing:* The robot speaks to the human, shares information such as the route direction and announces the next steps of the plan. The robot expects that they are paying attention to it. Therefore, we use the *Fulfilling robot expectations about social interaction* M_{Exp_SI} based on the attention ratio. Two parameters need to be defined for the scaling function, the bounds b_1 and b_2. As the minimum value for the metric, a ratio,

Metric id	Metric name	Metric equation – with Equations of Section 7.4	Scaled metric – with functions o
$M_{H\_contrib}$	Human contribution to the goal	nb_R_repet	$n_1(nb\_R\_repet) = 2 * \frac{nb\_R\_}{}$
M_{Exp_SI}	Fulfilling robot expectations about social interaction	$Ar = \frac{duration_{isAttentiveTo(robot)=true}}{duration_{robot\_speaks}}$	$n_1(Ar) = 2 * Ar -$
M_{DtG}	Distance-to-Goal	$\begin{cases} \Delta DtG(t=0) = 0 \\ \Delta DtG(t) = \max(0, \Delta DtG(t-1) - 1) \\ \quad \text{if } path\_length(t) < path\_length(t-1) \\ \Delta DtG(t) = \Delta DtG(t-1) + 1, \text{otherwise.} \end{cases}$	$-s_1(DtG(t)) = -1 + 2\exp\left(-\ln(2\right.$
M_{TtG}	Time-To-Goal	$\Delta TtG(t) = \max(0, e(t) + TtG(t) - TtG(T_0))$	$-s_1(TtG(t)) = -1 + 2\exp\left(-\ln(2\right.$

Table 8.5: Metrics used in the implementation presented in Section 8.7.

is 0 and the maximum value is 1, then $b_1 = 0$ and $b_2 = 1$. The QoI of the action is computed with this only metric.

(b) *Human-Robot Q/A process:* The robot asks a question to the human. As for the previous action, the robot expects the human to pay attention to it so we compute the QoI with M_{Exp_SI}. It also expects the human to give an appropriate answer. If it does not happen, it will ask the human to repeat, specifying that the answer has not been understood. We have limited the possible number of attempts to 3. After 3 attempts, the robot ends the task, as it cannot carry on with the task without an answer. So, we use *Human contribution to the goal* $M_{H\_contrib}$, the number of times the robot repeats. Because the maximal number of repetitions is 3, we set for the scaling function $b_1 = 3$ and $b_2 = 0$.

The QoI is computed with the two metrics: *Fulfilling robot expectations about social interaction* and *Human contribution to the goal*. The trials showed that the action QoI results were satisfying with the weights $W_i = 1, i \in [0, 1]$ as applying the Equation (7.1).

(c) *Ensuring that Human moves aside:* This action is used if, for pointing, the robot decides to place itself in a position which is very close to where the human is currently standing. In this case, the robot asks the human to step

aside to the right or left, depending on the human's future position. Then, we want to measure the progress of the human going further from the planned robot position. In order to do this, we use the *Distance-to-Goal* M_{DtG} but with the condition of the ΔDtG equation adapted, being if $path\_length(t) > path\_length(t-1)$ instead of if $path\_length(t) < path\_length(t-1)$. We scale the metric with $-s_1$, the additive inverse of the scaling function and not directly s_1 as the closer to 0 ΔDtG is, the better it is in terms of goal completion. From trials, we set $-s_1$ parameters values with $th = 5$ and $k = 1.5$.

If the human does not move or does not go far enough from the robot position, the robot will ask again with a limit of 3 trials (if the robot cannot move, it will carry on the task from their current positions). So, we use $M_{H\_contrib}$ as for the previous action.

(d) *Human-aware robot navigation:* The robot has to move from its initial position to its computed one. It navigates while respecting social constraints and its path may change as it adapts according to what the human is doing. At execution time, to measure the robot progress towards its goal, we use the *Time-to-goal* M_{TtG}, with the same scaling function than M_{DtG}. The QoI of the action is computed with this only metric.

(e) *Ensuring correct human placement for verbal interaction:* After it has moved, the robot asks the human to come in front of it. If the human is not perceived after a few seconds, the robot will ask again and so on in a maximum of 3 trials. If after these 3 times the human is still not perceived, the robot ends the task.

The QoI of this action is computed with $M_{H\_contrib}$ – we do not use M_{Exp_SI} as the human is not in the field of view when the robot is calling them.

(f) *Ensuring correct human placement for route explanation:* Once the human is in the robot field of view after the HR motion, they may not be at the right place to properly see what the robot has to point at. In this case, the robot will ask the human to move forward or backward according to what it has computed about the human perspective (*e.g.*, this is to avoid that an object occludes the view for the human). Then, we want to measure the human progress towards the position the robot has computed for them. In order to do this, we use the *Distance-to-Goal* M_{DtG}.

The robot stops giving instructions if it computes that the position of the human allows them to see the target, or after 3 trials, so we use $M_{H\_contrib}$. After 3 trials, if the human cannot see the target, still, the robot will carry on the task taking this into account.

Mall elements	Mockup mall	Real mall
Shops	19	140
Doors, stairs, elevators	10	50
Corridors	11	41
Levels	2	2

Table 8.6: Number of elements described in the mockup and real malls (geometric, topologic and semantic models in Fig. 8.5).

8.7.3 Proof-of-Concept

This section reports on an effective implementation of the approach as an illustrative proof of concept. We show the ability of the robot to conduct an interactive task, to assess in real-time the QoI and to track its evolution during three direction-giving task executions where the same human displayed a different way of behaving. In the three cases, the task was conducted until its end, in our lab where we reproduces the mall environment (Fig. 8.5a, Table 8.6). The computed QoI for each way is presented in Fig. 8.17. The three different ways of behaving are described in the following list:

- A human executed perfectly the expected actions and was not disturbing the robot when it navigated (*i.e.*, the "ideal" human from the robot point of view).
- A bit "confused" human tried to contribute to the task success but did not execute everything well. The human was, from time to time, not very attentive, as looking around. Also, they gave an answer to the first question that the robot did not understand, and then they took their time before answering again. Then, they prevented a bit the robot to move as it had planned and once the robot reached its position, they took time to come as close as the robot wanted.
- A human wanted to disturb the robot during the task. They gave three incomprehensible answers to the first question, blocked multiple times the robot in its move, waited for the robot to ask twice to come in front of it and finally asked the robot to point and explain the route three times.

Now, if we take a look at the QoI outputs of Fig. 8.17, we can see that their three shapes are very different. In Fig. 8.17a, we can observe that the task and actions QoIs remain with the highest value 1 all along. A graph as this one allows us to infer that everything went very smoothly during this direction-giving task. Then, we can guess that it corresponds to the execution performed with the 'ideal' human.

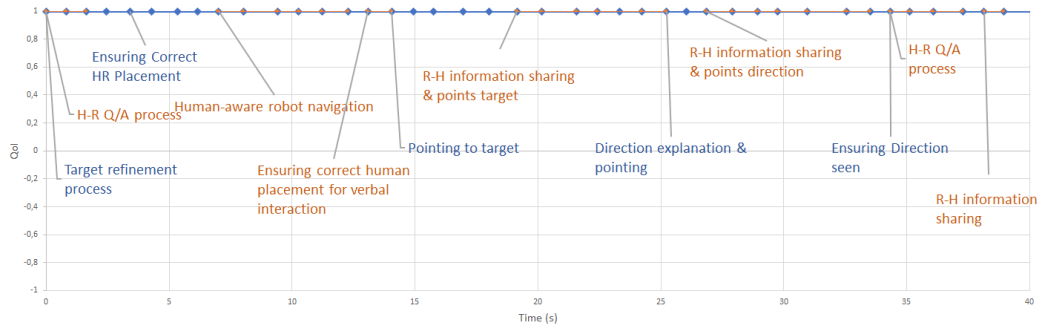
In Fig. 8.17b, we note that each subtask was executed in respect of the standard duration. If the QoI of *Target refinement process* drops it is because of the action QoI as the QoI of the *H-R Q/A process* drops because the robot had to repeat the

Action	QoI formula (metric aggregation)
Robot-Human information sharing	$M_{Exp\_SI}(t)$
Human-Robot Q/A process	$\frac{M_{Exp\_SI}(t) + M_{H\_contrib}(t)}{2}$
Ensuring that Human moves aside	$\frac{M_{DtG}(t) + M_{H\_contrib}(t)}{2}$
Human-aware robot navigation	$M_{TtG}(t)$
Ensuring correct human placement for verbal interaction	$M_{H\_contrib}(t)$
Ensuring correct human placement for route explanation	$\frac{M_{DtG}(t) + M_{H\_contrib}(t)}{2}$

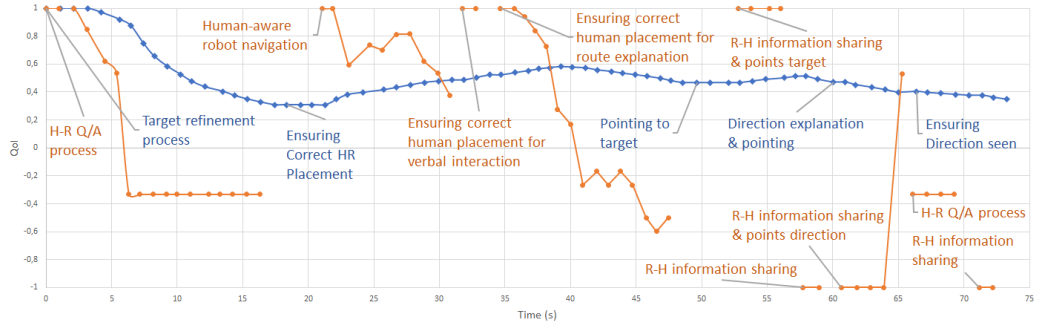
Table 8.7: QoI computation for each action as an aggregation of metrics

question and the human was not looking at it. From 21 seconds to 40 seconds, we can see the task QoI getting higher as the QoIs of *Human-aware robot navigation*, *Ensuring correct human placement for verbal interaction* and the beginning of *Ensuring correct human placement for route explanation* are quite high. Next, seeing the shape of the computed QoI of the action *Ensuring human placement for route explanation*, we can infer that the human was not moving as the robot wanted. Indeed, they took 10 seconds to make one step forward (they had 1 meter to cross). Because of that, the task QoI started to decrease again. In the final part of the task, the human was time to time attentive to the robot and answered quickly to the last question, so the task QoI remained rather equal with its final value being 0.34 which is above 0 so meaning a correct interaction.

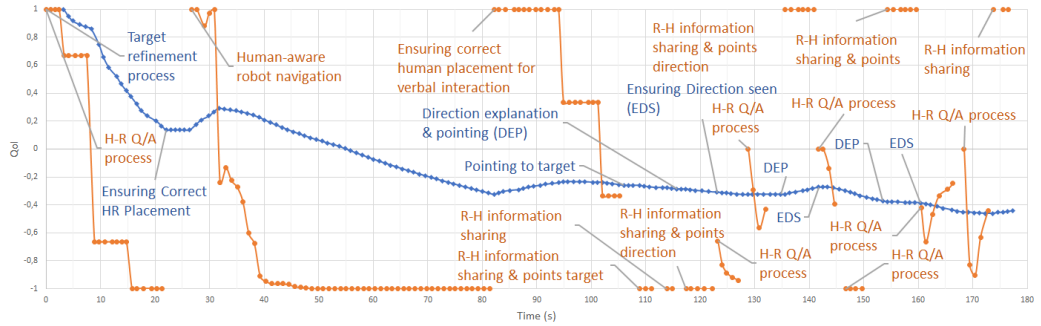
Finally, we can see in Fig. 8.17c that the final QoI of the task is -0.44 which allows us to infer that the task was not executed smoothly. And indeed, when we look at the shape of the task QoI, it only went down (or almost) all along the task. It is explained by some subtasks that took more time than they should have and also by some actions QoIs that are very low, especially the one of *Human-aware*



(a) Evolution over time of the measured QoI for the 'ideal' human. Both action and task QoIs remain at 1 as the task is proceeding smoothly.

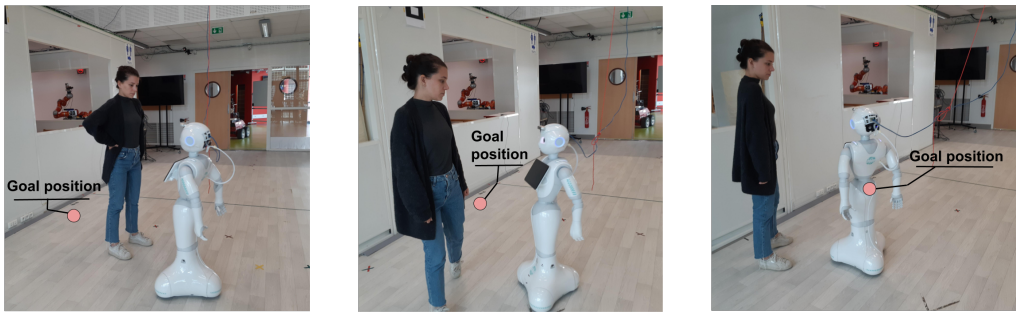


(b) Evolution over time of the measured QoI for the "confused" human. They took time to answer the first robot question and to move forward but the task QoI does not drop too much because the robot was able to give the route explanation without any issue even though the human was not very attentive.



(c) Evolution over time of the measured QoI for the non-compliant human. Several times the human did not give the expected answer to the robot during the target refinement process. Then, they blocked the robot path. After that, the robot had to ask twice the human to come in front of it. Finally, the robot repeated the route direction three times but still the human kept saying that they did not understand. Therefore, the task QoI decreases all along the task.

Figure 8.17: Evolution over time of the measured QoI for the route guidance task with three different human behaviors. The QoI for the task is drawn in blue, and the QoI for the actions is drawn in orange.



(a) Human who put themselves on the robot path, preventing the robot to navigate towards its goal position

(b) Human who put themselves on the robot path after it computed a new path to reach its goal position

(c) Human finally getting outside of the robot path, allowing it to reach its goal position

Figure 8.18: A human disturbing the robot during *Human-aware navigation*, preventing it to reach its goal position as planned.

robot navigation. At the beginning of the robot navigation, the estimated time to goal returned by the planner was 6 seconds but the robot actually took 50 seconds to reach its goal then the action QoI computed with $= M_{TtG}(t)$ was -1 for 40 seconds. And indeed, all along its navigation, the human was blocking the robot until they got tired of this game, as visible on Fig. 8.18.

In this example, we showed the QoI evaluation process integrated to a complete robotic architecture. The robot was able to assess the QoI in real-time while interacting with a human.

8.7.4 Discussion on the results of the QoI Evaluator

While a number of evaluation methods has been proposed to evaluate a human-robot interaction from the human perspective and often for analysis after performance, our choice to let the robot evaluate, on its own and in real-time the quality of its interaction with a human is quite new and original. To endow the robot with such an ability, we designed, implemented and tested a number of metrics and a method to aggregate them.

The work of Steinfeld *et al.* [278] was very helpful to design a first set of metrics and as an inspiration about what could be used. From there, we have elaborated and proposed a set of metrics which are meant to estimate of the quality of an ongoing interaction and not once it is over. The work of Hoffman regarding the *fluency* definition and how to measure it was also inspiring [130]. In a way, we extended his work by giving a meaning to the fluency measurement on the robot side, and in real-time – while their work applies to offline evaluation of shared workspace tasks. In Sect. 7.2, we mentioned systems measuring human affective states in real-time such as the framework developed by Tanevaska *et al.* [282]. Although we think such metric could be an interesting additional information to assess if an interaction is going well, we believe that these measurements do not offer an accuracy that

would lead to objective measurement of the quality of interaction, thus, we did not introduce them in our set for now. However, this could be done since our framework is designed to be open to new metrics. As for contributions, like the one proposed by Anzalone *et al.* [8], based on metrics such as gaze, head pose, body pose and response times to measure real-time engagement, we took them into account to some extent. However, the measure of the engagement that we propose should be refined depending on the inputs available on-line to the robot. Moreover, we will investigate how their work could be used in a more general way (*e.g.*, depending on the action that should be done and its context, human head pose and body posture could be a good indicator of effectiveness and not only engagement).

Our intention, when we developed the idea of the Quality of Interaction Evaluation, was to use such computation to feed the decision-making process of the robot and this is what we intend to do in the future. However, such framework can also be used to compare interactions between different humans and/or robots, eventually as a benchmark similarly to the work of Sanchez-Matilla [240] or as a way for developers to detect repetitive interaction issues with an unsupervised robot in a real-world environment.

As a proof-of-concept, we implemented and deployed a first version of a QoI Evaluator assessing task and actions QoIs. We tested it on an interactive robot dedicated to provide route guidance to customers in a large mall. The approach gave satisfactory results. It showed the potential ability of a robot to detect momentary decreases of the Quality of Interaction and also more serious degradation of it which may need drastic change of behavior for the robot. This is only a first step and it should be validated with a study where we will ask humans to evaluate the quality of their interaction with the robot in a similar manner. The goal will be to analyse and compare this to the evaluation of the interaction quality estimated by our robot and, based on that, investigate potential improvements.

Finally, we do not claim to have a perfect measure of the Quality of Interaction. However, although the concept of Quality of Interaction is quite abstract, Movellan *et al.* showed that when it is measured by human observers, the inter-observer reliability of the concept is quite high [207]. Therefore, we believe we can endow the robot with an effective and pertinent ability aiming at measuring the quality of an interaction. We are aware that the set of metrics we proposed to do so is not exhaustive but the framework is designed to be easily extended with new metrics.

8.8 User Study

inserer les resultats computer par Kathleen - en attente

Chapter 9

The Director Task: a Psychology-Inspired Task to Assess Cognitive and Interactive Robot Architectures

Contents

In this chapter, we propose a new psychology-inspired task, gathering perspective-taking, planning, knowledge representation with theory of mind, manipulation, and communication. Along with a precise description of the task allowing its replication, we present a cognitive robot architecture able to perform it in its nominal cases. In addition, we suggest some challenges and evaluations for the Human-Robot Interaction research community, all derived from this easy-to-replicate task.

The contribution presented in this chapter is excerpted from our work, published in the proceedings of the RO-MAN 2021 conference [243]. This contribution has been achieved in collaboration with other PhD students of the HRI teams, our mutual thinking leading to the formulation of this new task for HRI. Then, more specifically in relation to the software implementation, Guilhem Buisan was concerned about the task planning part. Guillaume Sarthou worked on the knowledge management. Kathleen Belhassein has designed the presented task with us giving her psychologist point of view to create a task on which user studies could be performed. The engineer Yannick Riou worked on the motion planning component allowing us to develop a task where the robot acts on its environment. My involvement in this task was on the supervision component. It was an evolved version of the one which ran for the direction-giving task presented in the previous chapter, *i.e.*, the JAHRVIS as described in Chapter 5. It has also been the opportunity to refine the architecture developed for the MuMMER project, leading to the one presented in this paper.

9.1 Introduction

Developing robotic architectures adapted to Human-Robot Interaction and thus able to carry out interactions in an acceptable way is still today a real challenge. The complexity comes, among other things, from the number of capabilities that the robot must be endowed with and therefore from the number of software components which must be integrated in a consistent manner. Such architectures should provide the robot with the capability to perceive its environment and its partners, to merge and interpret this perceptual information, to communicate about it, to plan tasks with its partner, to estimate the others' perspective and mental state, etc. Once developed, the evaluation of these architectures can be difficult because all these components grouped into a single system. The tasks we usually want the robot to handle must highlight a maximum of abilities, while still being simple enough to be reproduced by the community. Moreover, we should be able to conduct user studies with it to validate choices regarding naive users.

Since a long term goal of the robotic field is to see robots evolving in our daily life, many tasks and scenarios have been inspired by everyday activities. Even if these tasks offer a large variety of situation to be handle, since the human partner is not limited in his actions, they have the disadvantage of not highlighting some subtle abilities which are nevertheless necessary for good interaction. The robot guide task [247] in mall, museum, or airport, requires high communication skills to understand free queries (possibly involving chatting) and respond to them, whether to indicate a direction or to give advice. However, the perception needs can be limited due to the vast environments, as well as the perspective-taking needs due to the same perception of the environment by the robot and the human[1]. Finally,

[1]For sure we can find some tricky cases where it could help but they do not reflect common situations.

with such a task the human partner is not an actor of the task and just has to listen to the robot once their question is asked. Even if being in more constrained environments, bartender-like tasks [221] have the same disadvantages. Indeed, the human is considered as a customer, and as such, the interaction with the robot is limited. The robot will never ask the human to help it for performing a task and their actions do not require coordination either full collaboration.

To involve the human partner in the task and requiring him to act with the robot, assembly-like tasks [283] can be used. Nevertheless, in most cases, the human acts as an assistant rather than as a partner as full collaboration can be challenging to perform. The robot thus elaborates a plan and performs the assemble, then asks for help when detecting errors during the execution (*e.g.*, when it cannot reach some pieces). Here the task leads to unidirectional communication. Moreover, because in such a task both the robot and the human have equivalent knowledge about the environment, it can be hard to design situations where belief divergence appears and thus perspective-taking would be required.

Scaling down an everyday task to transform it into a toy task around a table can reduce the task complexity and allow easy reproducibility. Moreover, it allows the robot and the human to work in the vicinity of each other, with smaller robots for example. With the toy version of the assembly task presented in [45], the human is more involved in the task. They ask the robot to take pieces and to hold them to help them assemble a chair. Even if the communication is unidirectional, we could imagine inverting the roles to test different abilities. Moreover, communication implies objects referring with the use of various visual features about the entities. Even if both agents have the same knowledge about the environment, the communication is grounded according to the current state of the world. In this task, no decision has to be made by the robot but once again, inverting the roles could open other challenges.

To focus studies around perspective-taking and belief management, the Sally and Anne scenario, coming from a psychology test, has been studied in robotic [201]. In this scenario, the robot is an observer of a situation where two humans come and go from a room, and move an object from a box to another. Since a human is in the room when the other is acting, a belief divergence appears between the two humans and the robot has to understand it. While the task highlights the belief management, it is first limited regarding the perspective-taking since the human presence or not could be sufficient to estimate the humans beliefs[2]. Moreover, the humans do not act with the robot since it is just an observer of the scene. In addition, no goal is formulated and the human neither interacts with one another. Finally, no communication is needed in the task. The scenario is thus focussed on the analysis of a situation.

In this chapter, we first propose a new psychology-inspired task that we think to be challenging for the Human-Robot Interaction community and rich enough to

[2]When both humans are in the room they have the same perception of the scene but have different beliefs about hidden objects. Perspective-taking would be required if the humans could lean over the boxes to check what is inside.

be extended: the Director Task. Inter alia, it requires perspective-taking, planning, knowledge representation with theory of mind, manipulation, communication, and decision-making. Then, we present the robotic cognitive architecture that we develop to perform the task in its nominal cases. Finally, on the basis of the presented task and what has been developed, we present a discussion about the possible future challenges and evaluations for the research community, with possible extensions of the task.

9.2 The Director Task: From psychology to Human-Robot Interaction

In this section, we present the origins of the Director Task and the needs it aims to respond to regarding other tasks from the psychology. Then, we detail the setup we have designed in terms of objects characteristics and organization in the environment. We end this section with our adaptation and the required abilities we have identified.

9.2.1 The original task

The Director Task has been mainly used in psychology as a test of the Theory of Mind (ToM) usage in referential communication. This task originates from a referential communication game from Krauss and Glucksberg [170]. In this game, two participants are one in front of the other with an opaque panel between them. A speaker has to describe odd designs to a listener, either to number them for the adults or create a stack of cubes for the children. To refer to the odd figures, participants have to use images (*e.g.*, "it looks like a plane").

This game was then adapted by Keysar *et al.* [156] and became the Director Task. It has been used to study the influence of mutual knowledge in language comprehension. In this task, two people are placed one in front of the other but instead of an opaque panel between them, they place a vertical grid composed of different cells and objects in some of them. The **director**, a participant or in most cases an accomplice, instructs the **receiver**, a participant, about objects to move in the grid. The receiver thus follows the director's instructions about objects to move. The particularity of the task is that some cells are hidden from the director, meaning that the receiver, being on the other side of this grid, does not have the same perspective as the director. They thus know the content of more cells than the director and consequently sees more objects. When the director instructs the receiver to move an object, for a successful performance, participants must take the perspective of the director to move the right one. Because the configuration evolves all along with the task, they have to update their estimated perspective all along with the interaction.

For example in Figure 9.1, if the director asks for the smallest apple (*), the proper smallest (called competitor) is only visible by the participant and not by the

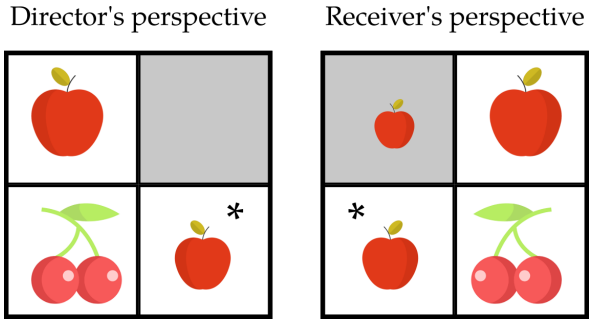


Figure 9.1: Sample display from the director's and the receiver's perspectives. The asterisk indicates the target object. Giving the sentence "the smallest apple" the receiver should find the good one even if he can see a smallest one in its perspective.

director. The participant then must understand the director's perspective to take the target apple and not the competitor one. Some studies showed that for their first attempt, participants considered or took the smallest apple from their own point of view and only after, the target one. These results were interpreted in various works as the participants understanding language in an egocentric way [154, 157, 155, 158]. Some social cognition studies used a computer-version of the Director Task whose results are consistent with the ones mentioned previously, namely that participants do not use ToM inferences in language interpretation [91].

Although they require the attribution of mental states to others, some authors have distinguished ToM tasks and perspective-taking tasks reporting distinct although related mechanisms. Santiesteban *et al.* [242] considered in their study that perspective-taking abilities were measured by the Director Task whereas ToM usage was investigated through another task called "strange stories" [121]. However, this ToM task requires the attribution of mental states to a story protagonist (to have knowledge of others' mental states), whereas the Director Task asks for adopting the perspective of the director in order to follow their instructions (to use this knowledge in order to execute the task properly). Thus, the authors estimated that the Director Task requires a higher degree of self-other distinction by continuously isolating our own perspective from the director one. In addition to perspective-taking abilities, the Director Task makes use of executive functions [237] and attentional resources [189].

The Director Task has thus been particularly used in psychology studies of referential communication, language comprehension, and perspective-taking abilities. However, to date it has never been exploited in the context of a HRI although this task presents interesting challenges for this field. It would not only bring technical challenges but also provide a way to investigate the different cognitive and behavioral processes involved in such a cooperative Human-Robot task.

9.2.2 The Director Task setup

The material used in this task has been chosen to be easily acquired and can be hand-built. It is composed of blocks, compartments, and a storage area. Each element is equipped with AR-tags allowing the robot to perceive them without advanced perception algorithms.

Director's perspective Receiver's perspective

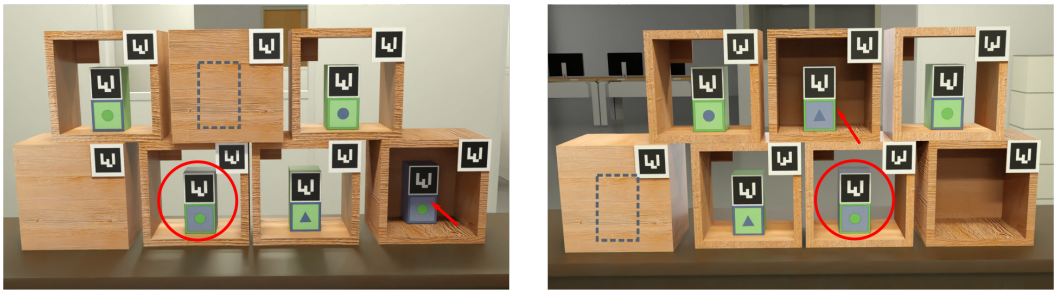


Figure 9.2: A director task setup adapted to the HRI with the director's and receiver's perspectives. For the material, each element (blocks and compartment) is equipped with AR-tags allowing their detection by the robot. Each block has four visual characteristics: a main color, a border color, a geometric figure, and a figure color. Compartments can be hidden for the director or the receiver. For the director to designate the block marked with a red circle, estimating the receiver's perspective, he can refer to it by its main color (blue) because he estimates the other blue block is not visible by the receiver. For the receiver, by taking into account the director's perspective, he can understand the referred block as he estimates the other blue block to not be visible by the director.

As shown in Figure 9.2, the blocks have a primary color covering them all. On two opposite faces, additional visual features are drawn. The top part of these faces is dedicated to the robot's perception with unique AR-tag on each face[3]. The bottom part is the same on both faces and is dedicated to the human perception with a main color, a border, and a geometric figure. Every visual feature (the colors and the forms) has exactly two variants. The colors are either blue or green and the figures are either a triangle or a circle. The figures and colors have been chosen in such a way to allow the emergence of "coded words" between the participant to identify a block. With a bit of imagination, some could refer to the left-most block through the sentence "the mountain in the sea" or the second leftmost by "the puddle". The number of features has been chosen to have sixteen block variants from which we remove the four uni-color variants (all the elements having the same color) to avoid too easy description of the kind "the fully green block". Regarding their description complexity, while the main color is directly related to a block, the other colors are respectively related to the border and the figure. This means

[3]because the tags are different on each side, the director can not refer to them as the receiver does not see the same ones

that for two blocks whose only difference is the color of one of these elements, the said element has to be referred to by its color. A description of a block involving all its features would be "the [color] block with the [color] border and the [color] [figure]". Such complete descriptions are hard for the human to process. In this way we expect the participants to minimize the complexity of their communication by referring to the blocks only using the features distinguishing them from other blocks.

Three types of compartment exist. Some are open on two of their opposite sides allowing both the receiver and director to see the content and to manipulate it. Some are open only on one of their sides meaning that only one of the participants can see and take what is inside. The other participant can thus neither know if a block is inside or not. The last compartment type has an open side and the opposite one equipped with a wire mesh. Because of the side with the wire mesh, both participants can see what is inside but only one of them can take it. With these three types, we will be able to test the impact of the awareness of the blocks (*e.g.*, a block is known to be present but not necessarily visible), the visibility of the blocks, and their reachability (*e.g.*, a block can be visible but not reachable).

Finally, one storage area, corresponding to the place where the receiver has to store the blocks, is delimited by a rectangle on a shelf.

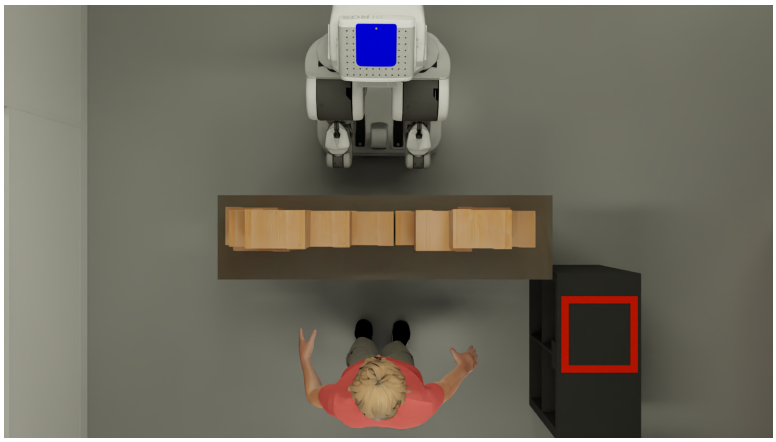


Figure 9.3: The Director Task setup with the robot and the human partner one in front of the other and a piece of furniture between them. Compartments are placed on top of the furniture and blocks are placed in the compartments. Next to the agent having the receiver role, here the human, a storage area is placed to drop the removed blocks.

Regarding the disposition, the compartments are stacked on a piece of furniture to create a kind of grid. The blocks can be put in a compartment. As illustrated in Figure 9.3, the two agents are placed one in front of the other with the furniture and thus the compartments between them. Finally, one storage area, corresponding to the place where the receiver has to store the blocks, is delimited by a rectangle on a shelf next to the receiver. In the figure, the human would be the receiver since

he has the storage area on his right.

9.2.3 The Director Task adaptation for HRI

In this section, we present the DT-HRI, the Director Task as we designed it for HRI, keeping the principle of two participants with a vertical grid between them. The high-level goal of the task is known by both agents: to put a set of blocks away. The precise goal is given by the experimenter to the director, either the robot or the human, i.e., the set of blocks that the receiver should remove from the compartments (see Figure 9.2).

As mentioned in the previous section, the Director Task characteristics bring a number of interesting challenges for a collaborative robot to solve. Because this is a task with two roles, one of the first challenges is to build a robotic architecture that gives the robot the ability to play both roles. Then, each role brings some problems to solve from a robotic point of view.

In the original task, the director knows they have a subset of the receiver's perspective, they can consider all the objects when communicating. Thus, only the receiver has to reason about the other's perspective, taking into account that some objects are not visible by the director. In order to enrich the task for HRI application, we propose to also have compartments hidden from the receiver and visible by the director (see Figure 9.2). Therefore, both roles have to perform perspective-taking, whether to give instructions or to understand them. On one hand, this challenging task allows to demonstrate the abilities of a robotic system. On the other hand, it is an easily reproducible scenario to perform user studies on human-robot interactions in a controlled environment.

To be able to study more specifically some skills, such as verbal communication, perspective-taking and adaptation, we defined a set of rules for both the robot and the participant. First, to focus the task on verbal communication, the agents are **not allowed to point** to objects, either with their hand or gaze. Then, to strengthen the perspective-taking aspect and not fall into a simple referential communication task, participants are **not allowed to use geometrical relations** in the verbal communications. They cannot, for example, say "the leftmost block" or "the block to the right of the green one". In this way they are limited to few visual features, with high ambiguity, therefore requiring to take into account the other perspective. Finally, to enable an evolution of the situation over time and thus requiring a constant adaptation during the interaction, the objects are not moved from one compartment to another but removed from the compartments. The **order of the instructions is free**, enabling the director to elaborate a strategy if needed.

9.2.4 A task to demonstrate the abilities of a robotic system

More than being an easily reproducible scenario to perform user studies on human-robot interactions in a controlled environment, the Director Task allows to demonstrate abilities of a robotic system. We detail here some abilities for which the task

has been designed for.

paragraphPerspective-taking abilities When working on the ToM in the HRI context, the Sally-Anne test has been used multiple times and allowed to demonstrate some systems [201]. But, one of the benefits of the Director Task compared to the Sally-Anne test is that the agents (human or robot/director and receiver) have not only to infer knowledge using the other's point of view but also to act so it is possible to acknowledge that they use it in decision-making.

Communication abilities Moreover, the task requires to put a focus on communications which is widely studied in HRI. Indeed, the communication about an object can be more or less efficient, depending on the number of characteristics given about the object or the pertinence of these characteristics (*e.g.*, in Figure 9.1, the director does not need to add "red" to "take the small apple" as there is no apple of a different color). The robot needs to be able to give proper instructions but also to understand the human ones.

Planning abilities When a large number of blocks has to be taken in the task goal, it quickly becomes complicated to communicate about some of them as the director would have to add a lot of adjectives to be able to refer to one block. Therefore when the robot is the director, it becomes interesting to integrate the communication and the task planning together. Indeed, depending on the order in which the blocks are designated, the complexity of instructions can decrease or increase. Then, the planner can return an optimal order in which the robot has to give the instructions to the human.

Contingencies handling abilities While performing the Director Task, errors can happen. Either because the director gives a wrong instruction or the receiver misunderstands the instruction and takes the wrong block. In both cases, it can be because of a wrong consideration of the other agent's perspective. In the latter case, the instruction might be right but hard to interpret by the receiver leading to an error from them. Finally, errors can happen because of a failed action execution (*e.g.*, a block falls on the floor), a system failure for the robot, inattention from the human, etc. A robot with a robust decision-making system will be able to analyze, try to determine their origin, and handle a number of these contingencies. For example, if the human takes the wrong block, the robot can react in different ways, *e.g.*, asking the human to put it back or saying nothing and re-planning if this block was among the ones to take. If errors happen repeatedly, the robot can react differently than for a punctual error.

9.3 The cognitive robot architecture

In this section, we present the architecture developed to handle the Director Task in its nominal case but also to allow for future extensions, endowing the robot with the

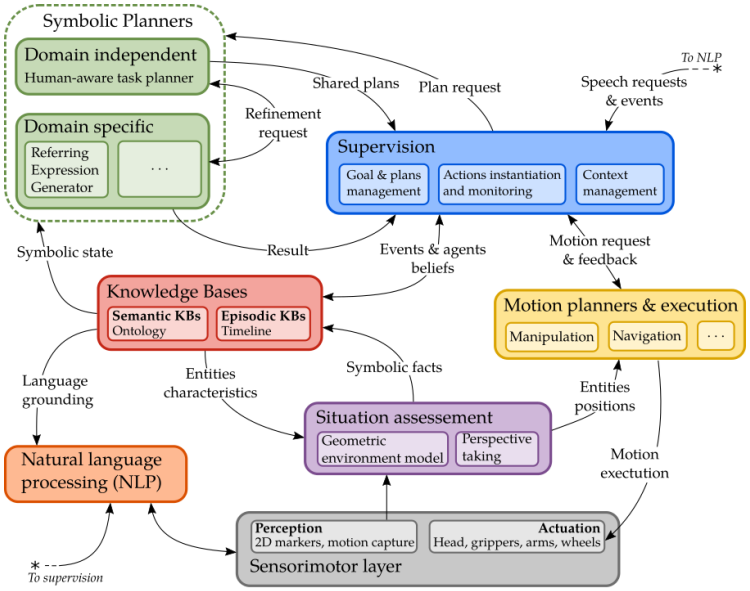


Figure 9.4: An overview of the architecture developed to handle the Director Task. Each block does not necessarily represent one software component but rather an architectural module (in terms of the features it implements). The arrows represent the type of information exchanged between the modules.

abilities described in section 9.2.4. The architecture is basically the one presented in Section 3.3. The seven identified modules are represented in Figure 9.4 with their respective communication links. In the rest of the section, we detail each module and how we have refined them in terms of functionality and linking.

9.3.1 Storing and reasoning on symbolic statements

As seen in the previous chapters, knowledge allows the robot to understand the environment it evolves in. Moreover, this same knowledge makes the robot able to communicate with its human partner about the current state of the world and ground the partner's utterance regarding this same world state.

Some have chosen to propagate their knowledge all along their architecture [122], each component enriching this knowledge at each stage. Others have preferred to see their knowledge base as an active server activating perception process regarding the searched information when needed [24].

As the architecture on which we based ours, we chose a central, server-based, knowledge base. We however refined it into two distinct sub-modules, the semantic knowledge base and the episodic one, as presented in Chapter 5, managed by Ontologenius.

9.3.2 Assessing the world: from geometry to symbolism

The role of the geometrical Situation Assessment module is first to gather different perceptual information and build an internal geometric representation of the world. From this world representation, the module then runs reasoners to interpret it in terms of symbolic statements between the objects themselves and between the involved agents and the objects. Doing so, the module only builds the robot's representation that does not necessarily reflect what the human partner believes about the world. This is the case with the occluded compartments. If a block is present in a compartment occluded from the human perspective, this block is not visible and thus unknown to the human and should not exist in their representation of the world. Here is the second role of our Situation Assessment module, estimating the human's perspective and building an estimation of their world representation. It is the first step allowing to implement the ToM principles (see Section 1.2).

To implement this module, we have chosen the Underworld framework [184]. It has the advantage to not be monolithic. Its principle is to create a set of worlds, each working at a different granularity and integrating specific features. It allows easy reuse of existing modules and makes the core reasoning capabilities independent of the used perception modalities. The worlds' structure we use is represented in Figure 9.5. At the top, there are the perception modalities, here AR-tags [95] for the objects and motion capture (mocap) system for the human detection. For each perception system, we define a world. In these worlds, we can filter the perception data depending on the system used. For the mocap, the data is clean enough. For the AR-tags we apply first a motion filter to discard data acquired when the robot moves and a field of view (FOV) filter to discard data from the border of the camera because of distortions. Moreover, both perception worlds can use the knowledge base presented previously to get the entities' CAD models and unique identifiers (UIDs) shared across all the components of the architecture. When the AR-tags world receives an AR id, it can query the semantic knowledge base to get the UID related to this tag and get its CAD model. As the output of these worlds, we ensure to have clean data with UID related to the knowledge base.

The world of the middle in Figure 9.5 is the robot's world representation. Information from the perception worlds is merged along with the static elements (the building walls) and the robot model. From there, geometric reasoners are applied to extract symbolic facts. In the current version of the system, the computed facts are *isOnTopOf*, for an object put on top of another, *isInside*, for a block in a compartment, *isVisibleBy*, assessing if an agent could see the object or not, and *isReachableBy*, assessing if an object can be taken by an agent. All these facts are sent to the robot's semantic knowledge base, where reasoners will deduce further facts. For example, if a block is in a compartment, the compartment has the block inside (inverse property), and if this compartment is on top of the table, the block inside is also above the table (chain axiom).

While the previous world corresponds to the robot's representation, the one below aims at representing the partner's one. From the previous world, we compute a

segmentation image from the human point of view and use it as a filtered perception world. This allows us to instantiate the same kind of world management process we used for the robot but this time for the human. In this way, we emulate their perception capability and the geometric reasoners can be run in the same way as previously. Symbolic facts are thus computed and sent to the human's semantic knowledge base. In the world of the bottom on Figure 9.5, we can see that the two blocks in the occluded compartments are not present in the human world. Here we make explicit the difference between an object that is unknown and an object that is known but not visible.

9.3.3 Planning with symbolic facts

The symbolic planners are divided into two categories: the domain-independent, planning high-level tasks, and the domain-dependant, specialized in solving precise problems. We first introduce the domain-specific ones and the domain-independent in a second time.

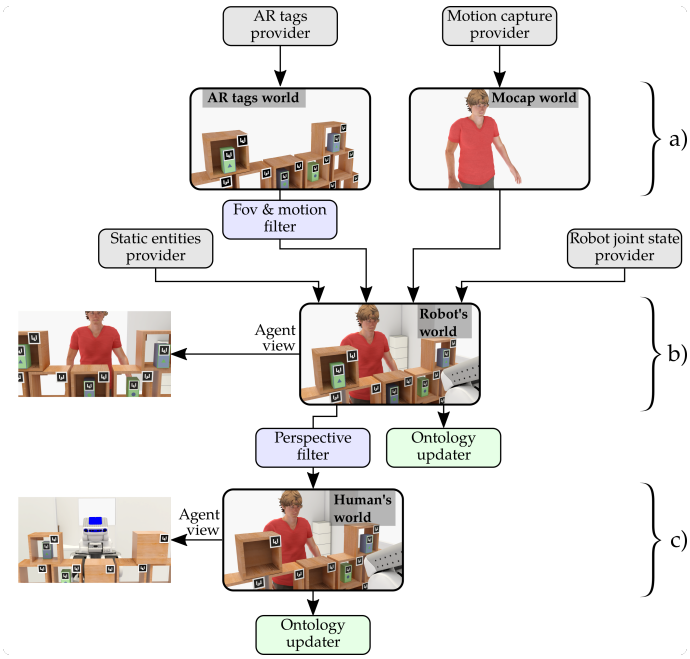


Figure 9.5: The world cascading structure of the geometrical situation assessment system. The two worlds at the top are build from the perception systems and filtered. The world of the middle merges the different perception information and computes symbolic facts on it. The world at the bottom is the estimation of the human world representation and is computed based on perspective-taking in the robot's world. Like for the world of the middle, symbolic facts are computed and sent to the semantic knowledge base.

9.3.3.1 Solving precise problems

Building a single monolithic planner could be an intractable challenge. Thus, we chose to consider a set of dedicated planners which could be reused from one system to another. In the current version of the system, only one specific planner has been identified. This planner is a Referring Expression Generator (REG) solver. Regarding the current symbolic state of the world, it aims at finding the minimal set of relations to communicate and allow the listener to identify a given entity. For example, wanting to refer to a block being the only one with a green triangle on it among the other, this planner can find that the only relations to communicate are the block's figure and the figure's color. With this information, the listener should be able to identify the referred block without ambiguity.

This planner is presented in [54] which is based on a Uniform Cost Search algorithm and which is to the date the most efficient one in term of computation time. It works with an ontology, being the semantic knowledge base presented previously. Because the communication information it generates will be interpreted by the robot's partner, we chose to give the estimated human knowledge base as input to the planner. Thanks to this, the blocks unknown from the human — i.e., hidden from them — are not taken into account as they cannot lead to any ambiguity for the listener. Moreover, this planner can take some constraints as input, related to the property usability and the context of the communication. The usable properties constraint prevents some properties to be used in a referring expression. Indeed, the input ontology is not dedicated to the specific referring expression generation problem and contains additional knowledge used by other modules as the objects' CAD models or tag UIDs, that does not aim to be communicated. The communication context aims at representing relations assumed to be already known by the listener. For the Director Task, when the robot asks the human to take a block, it assumes they know it is only talking about objects above the table around which the robot and the human are interacting. The already stored blocks — not on the table anymore — are thus not taken into account in the communication. If needed the communication context can be refined, for example by defining that the robot — and thus should the human as well — will only consider visible blocks and reachable blocks.

9.3.3.2 Planning for self and others

In the context of a Human-Robot interaction, when planning how to perform a high-level task, one has to take into account the human's contribution. Our current task planner is HATP/EHDA mentioned in Section 6.4. This planner allows the robot to plan by emulating the human decision, action, and reaction processes. For the Director Task, emulating the human reaction to a given instruction enables the comparison between multiple blocks order, the communication of higher-level instructions to the human (*e.g.*, ask to withdraw rather than take then put down) and the balance between multiple communication modalities.

As at execution time the supervision uses the REG, a domain-specific planner, and because the task planner uses the same type of knowledge representation, thus HATP/EHDA can use this planner during its planning process. In the current architecture, it can thus estimate the cost and the feasibility of referring communication by calling the REG.

9.3.4 Managing the interaction

Based on the components presented above, JAHRVIS manages the execution of Director Tasks, based on its processes presented in Chapter 5.

9.4 Experiments

modifs pour axer supervision-texte de guillaume

The architecture has been successfully implemented on a PR2 robotic platform. The robot is thus able to play both roles, the director and the receiver. In this section, we comment and analyze a video[4] of two experiments. For both experiments, the initial state is the same and are represented in Figure 9.6. The only emulated element is the human action recognition to trigger the next actions of the robot when it holds the director role as at the time, the Human Actions Recognition (HAR) process presented in Section 6.3 was not developed yet.

Figure 9.6: Initial configuration for both case studies. The top-right block is not visible, and thus unknown, by the human partner. The robot can not know if there is a block in the bottom-left compartment. All others blocks are known by both the robot and the human.

[4] https://youtu.be/jtSyZeqBkp0

9.4.1 PR2 as the director

We start this section with a PR2 in the role of the director (0:21 in the video). The setup is composed of six compartments including two compartments with a hidden face. One of these compartments is hidden from the human (the receiver) and one from the robot (the director). One block has been placed in each compartment. Consequently, only four blocks are known by both the human and the robot. Figure 9.7 is a visualization of the estimated geometric world of the human, maintained by the situation assessment component. Even if a block is present in each compartment, the leftmost one is not present in the estimation of the human's world. This absence comes from the fact that the human can not see what is in the compartment and thus can not know this block.

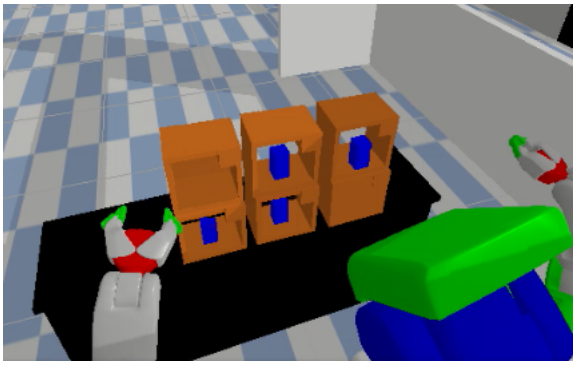


Figure 9.7: A visualization of the human's estimated geometric world from a third-person view. Even if a block is present in each compartment, the right most one is not present in this worls since the human can not see this block.

Figure 9.8 represents the entire interaction when the robot is the director. At the initial state, four blocks are visible from both agents. Describing them with all their visual features, they are:

- A blue block with a blue border and a green triangle
- A blue block with a blue border and a green circle
- A blue block with a green border and a blue triangle
- A green block with a green border and a blue circle

Thanks to the estimation of the communication cost at task planning using the results of the REG, the robot is able to find the optimal sequence of blocks to instruct. The overall communication is thus minimized and the RE is unambiguous in each situation. In the initial state (a to b), the robot asks for the green block as only one of the visible blocks is green. Since the green block has a circle on it, removing it, only one of the remaining blocks has a circle on it. The robot can thus use this feature to refer to the next block (b to c). Without communication cost

estimation during the task planning, such a simple situation would not necessarily appear.

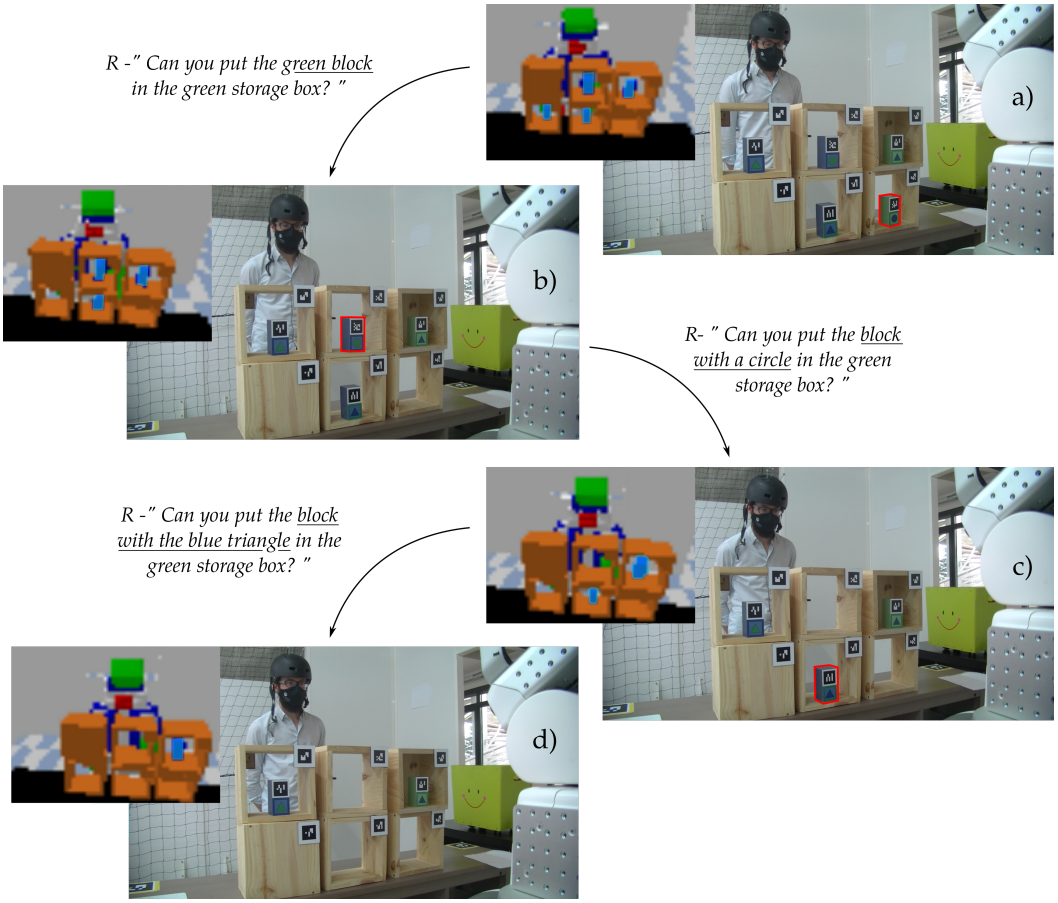


Figure 9.8: The director task handled by an autonomous PR2 robot in the role of the director. Each picture represents a step toward the achievement of the task. The estimated human perspective is displayed in the top left-hand corner of each picture. On top of the arrows leading to a new state are the sentences said by the robot to the human. The block outlined in red are the blocks referred to at each step.

9.4.2 PR2 as the receiver

While in its previous role the robot just had to instruct the human, when the robot is the receiver (1:33 in the video) more reasoning is needed. A retranscription of part of the interaction is represented in Figure 9.9. In the initial state, the same four blocks as previously are visible by both the agents. The robot is able to understand three actions: take, drop, and remove. The latter action is a combination of the two others.

For the first block (a to b on the figure), the human instructs the robot for the green block. The natural language understanding module returns the SPARQL

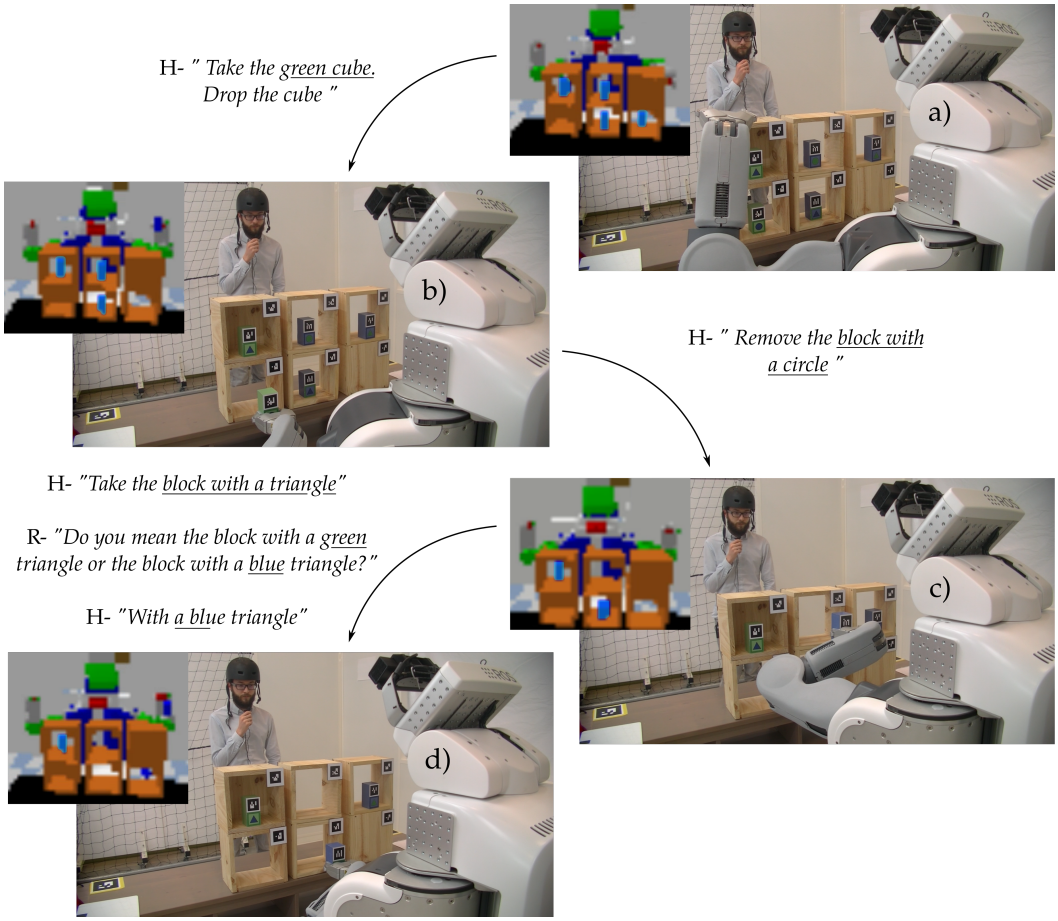


Figure 9.9: The director task is handled by an autonomous PR2 robot in the role of the receiver. Each picture represents a step toward the achievement of the task. The estimated human perspective is displayed in the top left-hand corner of each picture. On top of the arrows leading to a new state are the sentences said by the human to the robot and for the last situation the refinement query from the robot to the human, followed by the answer of the human.

query:

(?0, isA, Block), (?0, hasColor, green)

Since the robot assumes the human to speak about objects on the table, the understood query is merged with another one representing the context of the task: (?0, isAbove, table_1). Querying the human estimated ontology with the merged query, only one entity match. There is no ambiguity in human instruction. The robot takes the instructed block then drop it. If the query was applied to the robot ontology, two blocks would have matched since the block unknown by the human is also green. It goes the same for, the second instruction. There is no ambiguity. The SPARQL query related to this second block is:

(?0, isA, Block), (?0, hasFigure, ?1), (?1, isA, Circle)

The third instruction given by the human as the director is the most interesting for us. The human asks for *"The block with a triangle"*. However, the speech to text returns *"take is about to whip a triangle"*. With this sentence, the NLU module can only extract two known concepts being "take" and "triangle". Due to the limited amount of words understood, it does not try to generate a SPARQL query. The robot thus informs the human about its incapacity and repeat the heard sentence as a back loop for the human. At the second try, the sentence is understood and gives the query:

(?0, isA, Block), (?0, hasFigure, ?1), (?1, isA, Triangle)

However, matching this query to the human's estimated ontology, we get two results. Once again, matching it to the robot's ontology would give three results but the third one is not visible from the human. Since all the concepts of the sentence have been understood and linked together to create the query, the human should have made a mistake, providing an ambiguous referring expression.

To be proactive, we want the robot to ask precision about the block to take by proposing visual features to distinguish them. To do so, we use the REG algorithm on each ambiguous block. As a context for the REG, we pass the previously merged SPARQL query. It represents what has already been understood by the robot. In the current situation, the robot thus performs two REG and their results are used to generate the disambiguation sentence:

"Do you mean the block with a green triangle or the block with a blue triangle?"

When the human responds, for sure it does not generate a complete description f the block to be taken. It rather answers the question. The query extracted from his answer is thus combined with the previously understood one in case some information is missing. Matching this last query to the human's estimated ontology, the robot finally get the block to remove.

With this latter case, we saw how the robot can react to a human's mistake and use the REG to help the progress of the task, even if it is the receiver.

9.5 Open challenges for the community

So far, we proposed a cognitive robot architecture handling the Director Task in its simplest form, both roles. In this section, we now present some open challenges for the community around the task. Moreover, because the task can be performed in a controlled environment, we also present in a second part some user studies to investigate ways of sharing information.

9.5.1 Some challenges to take up

Challenged abilities / components	Challenges
Perspective-taking	1
Communication	4, 6, 7
Task planning	2, 3, 4
Reference generation	4, 5, 8, 9
Contingencies handling	1, 2, 3, 4

1. Fine contingency analysis: Due to the high ambiguity between the blocks and the presence of occluded compartments, failures can easily arise and have to be handled. In the case the human is the receiver and does not take the instructed block, the robot has to determine the origin of the failure. It could come from a perspective not taken into account either by the director or the receiver, a block description not clear enough, or just an error of the receiver regarding a correct (non-ambiguous) description.
2. Not handling contingencies as errors: Based on the example of the previous challenge, the human takes another block than the one instructed but this block could be part of the next ones to take in the plan. In this case, why the robot should try to "repair" i.e., make the human takes the instructed block? Maybe it could mention to them that they took the wrong block but it does not matter because this one is also part of the plan. Then, either the robot could re-plan or even better, use a conditional plan and adapt according to the human's actions.
3. Handling errors as errors: Still based on the case where the human takes another block than the one instructed, the robot has to communicate and negotiate with them in order, first to fix the error i.e., put back the block to its original compartment, then adapt its original instruction to make it clearer and improve the chances to have them take the right one.
4. Changing something when recurrent failures: In case of recurrent failures by the partner, during one interaction session (multiple tasks can be performed in one session) or along with several ones, the robot could try to analyse the origin of the failures and adapt itself to increase the QoI and reduce the failures. It could be through properties' cost adaptation if the partner has some difficulties with certain visual features or communication context adaptation if the partner took the stored blocks into account in its understanding.
5. Allowing spatial references: As explained in section 9.2.1, the Director Task is originally a task to test referential communication. Even if the present version asks the participants to not use spatial reference, this rule could be relaxed to study perspective-corrected spatial Referring Expression Generation.
6. Understanding the human instructions: In the current implemented version, the robot can only understand a precise vocabulary, being the one describing the blocks in the way we have thought them. In a more natural interaction, humans could use a richer vocabulary, give instruction in multiple steps or have communications not directly linked to the task. During tests for designing the

task, it was common to have instructions like "take the block with a ... triangle. No, rather the one with a green border". Such complex communications should have to be managed by the robot.

7. Introducing code words: As presented in Section 9.2.2, the visual features on the blocks have been designed to be able to see landscapes on them, with a little imagination. During the interaction, the robot could thus try to negotiate some coded words in order to be more efficient in the task considering multiple sessions. Being the receiver, it would have to understand the coded words as to be part of a description and remember them.
8. Referring to a past event: When a human performs multiple times the Director Task with the robot, noteworthy events can happen. These events could be recognized and recorded by the robot so it can refer to them when speaking about an object (*e.g.*, "can you take the one you dropped in the previous task ?"). Likewise, the human may also use these past events and the robot would have to understand them.
9. Communicating about multiple blocks in a raw: Instead of giving instructions one at a time, the director could give instructions for multiple blocks in a raw. This may bring different kinds of communications from the base task as "I do not remember the instruction for the last block" when the human is the receiver. Also, this would be interesting for planning when the robot is the director as it could give instructions such as "Take all the blocks with a triangle on them" and it would be a different kind of instructions to interpret when the robot is the receiver.

9.5.2 Some Director Task-based user studies to perform

Some robot behaviours, mainly about the referring expression generation, have been designed with regard to the current literature but could be refined thanks to user studies based on the Director Task. The references to the blocks involve the minimum of visual features allowing to discriminate them without ambiguity to fit the Grice's Maxim of Quantity [115]. However, due to all the cognitive mechanisms to use in this task (*e.g.*, perspective-taking) and the high ambiguity among the blocks, evaluating such behaviour compared to a full explanation could be interesting. Indeed, giving a reference with more information than needed would ensure to not match blocks being only visible by the receiver, which could help them to select the right block.

As presented in section 9.2.2, a special compartment equipped with a wire mesh can be used. Referring to a block matching also the one in this particular compartment could disturb the receiver or at least require a higher cognitive load to determine the right block to take. Such behavior could also be interesting to evaluate. In the same way, a block that was visible by the receiver and that the director move in a hidden compartment could disturb the receiver.

Evaluating such behaviours in a controlled task where the participants cannot know the real goal of the study could help the community in the design of architec-

tures applied to more realistic scenarios.

Conclusion

On the Human Agent Interaction Guidelines

Limitations and Future Work

Notes

Appendix A

Code of JAHRVIS ROS-Jason Agents

utile ? ajouter les autres ?

add colors and rules

A.1 Human Actions Monitoring

```
+NewPredicate[source(percept)]
   :  (action(_30, _31, Name, _32, Params)[source(robot_executor)] &
      isPredicateRobotAction(NewPredicate, Params))

+NewPredicate[source(percept)]
   :  (isMovementRelatedToActions(NewPredicate,
      ActionList) & (possibleStartedActions(ActionList2) &
      (jia.intersection_literal_list(ActionList, ActionList2, I) &
      ((.length(I)) > 0))))
   <- jia.update_action_list(possibleStartedActions, ActionList);
      ++possibleStartedActions(ActionList).

+NewPredicate[source(percept)]
   :  isMovementRelatedToActions(NewPredicate, ActionList)
   <- ++possibleStartedActions(ActionList).

+NewPredicate[source(percept)]
   :  (isProgressionEffect(NewPredicate, ActionList) &
      matchingStartedActions(Predicate, ActionList, ActionList2))
   <- jia.update_action_list(possibleStartedActions, ActionList2);
      ++possibleProgressingActions(ActionList2).

+NewPredicate[source(percept)]
   :  isProgressionEffect(NewPredicate, ActionList)
   <- ++possibleProgressingActions(ActionList).

+NewPredicate[source(percept)]
   :  (isNecessaryEffect(NewPredicate, ActionList) &
      (matchingProgressingActions(Predicate, ActionList,
      ActionList2) | matchingStartedActions(Predicate, ActionList,
      ActionList2)))
```

```
   <- !addFinishedActions(ActionList2).

+!addFinishedActions(ActionList)
   <- jia.update_action_list(possibleStartedActions, ActionList);
      jia.update_action_list(possibleProgressingActions,
      ActionList);
      ++possibleFinishedActions(ActionList).

+NewPredicate[source(percept)]
   :  isNecessaryEffect(NewPredicate, ActionList2)
   <- ++possibleFinishedActions(ActionList2).

+possibleStartedActions(ActionList)
   <- .send(human_management, tell, possibleStartedActions(ActionList));
      .fork(or, .wait((possibleProgressingActions(A) &
      (jia.intersection_literal_list(A, ActionList, I) &
      ((.length(I)) > 0)))), .wait((possibleFinishedActions(A)
      & (jia.intersection_literal_list(A, ActionList, I) &
      ((.length(I)) > 0)))), !timeoutMovement(ActionList)).

+!timeoutMovement(ActionList)
   <- .wait(50000);
      -possibleStartedActions(ActionList)[add_time(_33)].

+possibleProgressingActions(ActionList)
   <- .send(human_management, tell, possibleProgressingActions(ActionList));
      .fork(or, .wait((possibleFinishedActions(A) &
      (jia.intersection_literal_list(A, ActionList, I) &
      ((.length(I)) > 0)))), !timeoutProgressing(ActionList)).

+possibleFinishedActions(ActionList)
   <- .send(human_management, tell, possibleFinishedActions(ActionList));
      -possibleFinishedActions(ActionList).

-possibleStartedActions(ActionList)
   <- .send(human_management, untell, possibleStartedActions(ActionList)).

-possibleProgressingActions(ActionList)
   <- .send(human_management, untell, possibleProgressingActions(ActionList)).

+!timeoutProgressing(ActionList)
```

```
   <- .wait(30000);
      -possibleProgressingActions(ActionList)[add_time(_34)];
      .findall(possibleStartedActions(A), (possibleStartedActions(A)
      & (jia.intersection_literal_list(A, ActionList, I) &
      ((.length(I)) > 0))), L);
      .foreach(.member(M, L), -M;
      +M).

+!drop(G)
   <- !reset.
```

Appendix B

Scaling Functions

As the metrics are aggregated to compute the QoI, their values need to be on the same scale. In order to do this, we use scaling functions rescaling metrics into a range of $[-1, 1]$, as the QoI bounds. As all the metrics does not have the same properties, they have to be scaled by using different functions. The two properties to check to choose which function to apply to which metric are the following ones:

- does the metric already have a bounded value ?
- what value of the metric should make the QoI decrease, increase or remain the same ?

Therefore, we designed three functions to be used with metrics having bounded values and three functions for metrics that do not have upper bounds. Then, among these two sets of functions, it is possible to choose the one to use according to the positive, neutral or negative impact a value should have on the QoI.

B.1 Scaling of bounded metrics: Min-Max Normalization

We defined three min-max normalization functions, illustrated in Fig. B.1. They were designed to be used for metrics whose values belong to a bounded set, *i.e.*, metrics for which the minimum and maximum values are known. The first function is to apply in cases for which a measure approaching the bound value b_1 has a negative impact on the quality evaluation whereas a measure approaching b_2 has a positive one. It allows to scale a measure x between -1 and 1:

$$n_1(x) = 2 * \frac{x - b_1}{b_2 - b_1} - 1 \tag{B.1}$$

The second function is intended to be applied in cases for which a measure approaching the bound value b_1 has a neutral impact on the quality evaluation whereas a measure approaching b_2 has a positive one. It allows to scale a measure x between 0 and 1:

$$n_2(x) = \frac{x - b_1}{b_2 - b_1} \tag{B.2}$$

Finally, the last function is to apply in cases for which a measure approaching the bound value b_1 has an negative impact on the quality evaluation whereas a measure

approaching b_2 has a neutral one. It allows to scale a measure x between -1 and 0:

$$n_3(x) = \frac{x - b_2}{b_2 - b_1} \tag{B.3}$$

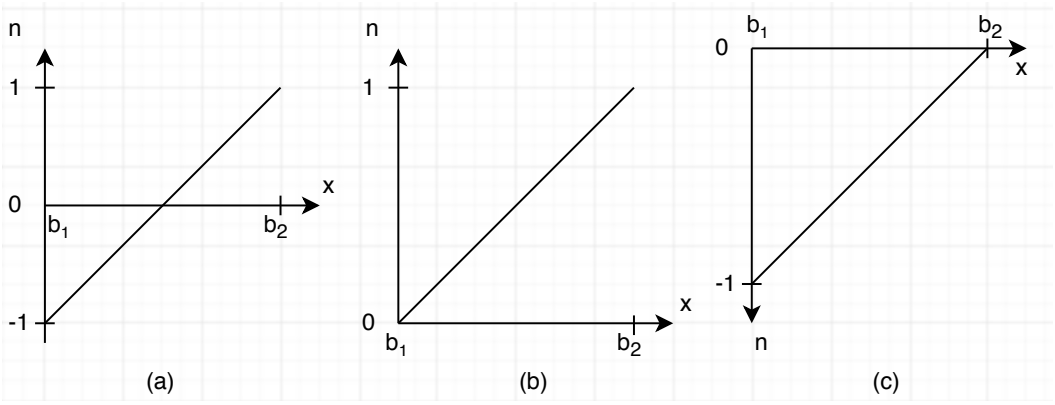


Figure B.1: (a), (b) and (c) respectively represent the min-max normalization functions (B.1), (B.2) and (B.3)

B.2 Scaling of unbounded metrics: Sigmoid Normalization

We defined three sigmoid-like functions to scale and squash values of metrics without an upper bound. As for the min-max normalization, there is one function to scale the metrics values between -1 and 1, another one to scale between 0 and 1 and the last one to scale between -1 and 0.

The first function allows to scale between -1 and 1 the values of a metric, for a metric whose values are between 0 and $+\infty$ (e.g. a duration whose final value is unknown during the execution). The function is defined as:

$$s_1(x) = 1 - 2\exp\left(-\ln(2)\left(\frac{x}{th}\right)^k\right), x > 0 \tag{B.4}$$

with $s_1(x) \in [-1, 1]$, th the value of the sigmoid's midpoint (*i.e.*, $s_1(th) = 0$) and, k setting the shape of the function curve. k and th values are set off-line by the designer and they allow to define the shape of the metric scaling.

The second function is designed for metric which cannot have a negative impact on the QoI as it scales the value between 0 and 1 (and with $x \in [0, +\infty]$ as well):

$$s_2(x) = 1 - \exp\left(-\ln(2)\left(\frac{x}{th}\right)^k\right), x > 0 \tag{B.5}$$

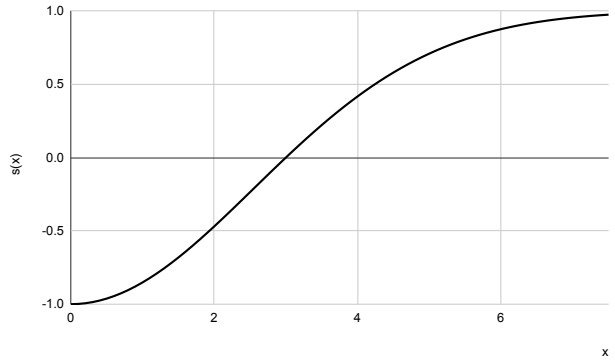
with $s_2(x) \in [0, 1]$, th the value of the sigmoid's midpoint (*i.e.*, $s_2(th) = 0.5$) and, k setting the shape of the function curve.

The third function is designed for metric which cannot have a positive impact on the QoI as it scales the value between -1 and 0 (and with $x \in [0, +\infty]$ as well):

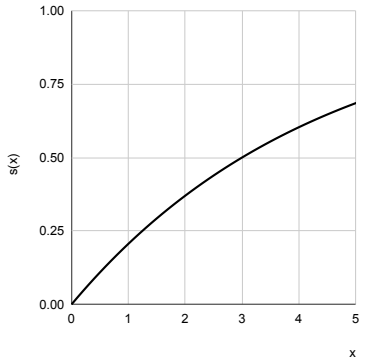
$$s_3(x) = -1 + \exp\left(-\ln(2)\left(\frac{x}{th}\right)^k\right), x > 0 \tag{B.6}$$

with $s_3(x) \in [-1, 0]$, th the value of the sigmoid's midpoint (*i.e.*, $s_3(th) = -0.5$) and, k setting the shape of the function curve.

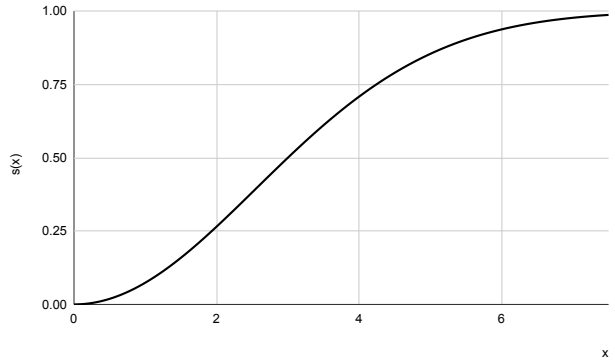
The functions $s_1(x)$ and $s_2(x)$ are illustrated in Fig. B.2 with four examples.



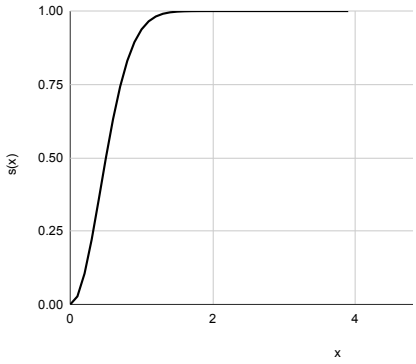
(a) Plot of $s_1(x)$ with $th = 3$ and $k = 2$



(b) Plot of $s_2(x)$ with $th = 3$ and $k = 1$



(c) Plot of $s_2(x)$ with $th = 3$ and $k = 2$



(d) Plot of $s_2(x)$ with $th = 0.5$ and $k = 2$

Figure B.2: Plots of the sigmoid-like functions $s_1(x)$ and $s_2(x)$ with different parameters values

Bibliography

[1] Norman Donald A. *The psychology of everyday things.* Basic Books, New York, 1988. (Cited in page 19.)

[2] Henny Admoni and Brian Scassellati. Social eye gaze in human-robot interaction: a review. *Journal of Human-Robot Interaction*, 6(1):25–63, 2017. (Cited in page 33.)

[3] Rachid Alami, Aurélie Clodic, Vincent Montreuil, Emrah Akin Sisbot, and Raja Chatila. Toward human-aware robot task planning. In *AAAI spring symposium: to boldly go where no human-robot team has gone before*, pages 39–46, 2006. (Cited in page 33.)

[4] James S Albus. Outline for a theory of intelligence. *IEEE transactions on systems, man, and cybernetics*, 21(3):473–509, 1991. (Cited in page 50.)

[5] Gary L. Allen. Gestures accompanying verbal route directions: Do they point to a new avenue for examining spatial representations? *Spatial Cognition & Computation*, 3(4):259–268, 2003. (Cited in page 135.)

[6] Federica Amici and Lucas M Bietti. Coordination, collaboration and cooperation: Interdisciplinary perspectives. *Interaction Studies*, 16(3):vii–xii, 2015. (Cited in page 11.)

[7] Kristin Andrews. *Do apes read minds?: Toward a new folk psychology.* MIT Press, 2012. (Cited in page 23.)

[8] Salvatore M. Anzalone, Sofiane Boucenna, Serena Ivaldi, and Mohamed Chetouani. Evaluating the engagement with social robots. *International Journal of Social Robotics*, vol. 7(4):pp. 465–478, Aug 2015. (Cited in pages 117 and 170.)

[9] Michael Argyle. *Social Interaction.* Transaction Publishers, 1973. (Cited in page 6.)

[10] Janet Wilde Astington and Jennifer M Jenkins. Theory of mind development and social understanding. *Cognition & Emotion*, 9(2-3):151–165, 1995. (Cited in page 9.)

[11] John Langshaw Austin. *How to Do Things with Words.* Clarendon Press, 1962. (Cited in page 23.)

[12] Patric Bach, Toby Nicholson, and Matthew Hudson. The affordance-matching hypothesis: how objects guide action understanding and prediction. *Frontiers in human neuroscience*, 8:254, 2014. (Cited in page 19.)

[13] Roger Bakeman and Lauren B Adamson. Coordinating attention to people and objects in mother-infant and peer-infant interaction. *Child development*, pages 1278–1289, 1984. (Cited in page 20.)

[14] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh P. N. Rao, and Minoru Asada. Efficient human-robot collaboration: When should a robot take initiative? *International Journal of Robotics Research*, vol. 36(5-7):pp. 563–579, 2017. (Cited in page 115.)

[15] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh PN Rao, and Minoru Asada. Efficient human-robot collaboration: When should a robot take initiative? *The International Journal of Robotics Research*, 36(5-7):563–579, 2017. (Cited in page 48.)

[16] Yvonne Barnes-Holmes, Louise McHugh, and Dermot Barnes-Holmes. Perspective-taking and theory of mind: A relational frame account. *The Behavior Analyst Today*, 5(1):15, 2004. (Cited in page 9.)

[17] Simon Baron-Cohen. *Mindblindness*. MIT Press, 1995. (Cited in page 34.)

[18] Simon Baron-Cohen, Alan M Leslie, and Uta Frith. Does the autistic child have a "theory of mind"? *Cognition*, 21(1):37–46, 1985. (Cited in page 10.)

[19] Pablo Barros, Nestor T Maciel-Junior, Bruno JT Fernandes, Byron LD Bezerra, and Sergio MM Fernandes. A dynamic gesture recognition and prediction system using the convexity approach. *Computer Vision and Image Understanding*, 155:139–149, 2017. (Cited in page 34.)

[20] Elizabeth Bates. *The emergence of symbols: Cognition and communication in infancy*. Academic Press, 1979. (Cited in page 24.)

[21] Andrea Bauer, Klaas Klasing, Tingting Xu, Stefan Sosnowski, Georgios Lidoris, Quirin Muhlbauer, Tinguang Zhang, Florian Rohrmuller, Dirk Wollherr, Kolja Kuhnlenz, et al. The autonomous city explorer project. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1595–1596. IEEE, 2009. (Cited in page 132.)

[22] Andrea Bauer, Dirk Wollherr, and Martin Buss. Human–robot collaboration: a survey. *International Journal of Humanoid Robotics*, 5(01):47–66, 2008. (Cited in page 67.)

[23] Cristina Becchio, Luisa Sartori, and Umberto Castiello. Toward you: The social side of actions. *Current Directions in Psychological Science*, 19(3):183–188, 2010. (Cited in page 11.)

[24] Michael Beetz, Daniel Beßler, Andrei Haidu, Mihai Pomarlan, Asil Kaan Bozcuoğlu, and Georg Bartels. Know rob 2.0—a 2nd generation knowledge

processing framework for cognition-enabled robotic agents. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2018. (Cited in page 180.)

[25] Michael Beetz, Lorenz Mösenlechner, and Moritz Tenorth. Cram—a cognitive robot abstract machine for everyday manipulation in human environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1012–1017. IEEE, 2010. (Cited in pages 40 and 50.)

[26] Esubalew Bekele and Nilanjan Sarkar. Psychophysiological feedback for adaptive human–robot interaction (hri). In Stephen H. Fairclough and Kiel Gilleade, editors, *Advances in Physiological Computing*, pages pp. 141–167. Springer London, 2014. (Cited in page 117.)

[27] Kathleen Belhassein, Aurélie Clodic, Hélene Cochet, Marketta Niemelä, Päivi Heikkilä, Hanna Lammi, and Antti Tammela. Human-human guidance study. Technical report, 2017. (Cited in page 135.)

[28] Kathleen Belhassein, Víctor Fernández Castro, and Amandine Mayima. A horizontal approach to communication for human-robot joint action: Towards situated and sustainable robotics. In *Culturally Sustainable Social Robotics*, pages 204–214. IOS Press, 2020. (Cited in page 10.)

[29] Kathleen Belhassein, Víctor Fernández Castro, Amandine Mayima, Aurélie Clodic, Elisabeth Pacherie, Michèle Guidetti, Rachid Alami, and Hélène Cochet. Addressing joint action challenges in hri: Insights from psychology and philosophy. *Acta Psychologica*. submitted. (Cited in page 10.)

[30] S. Bensch., A. Jevtić., and T. Hellström. On interaction quality in human-robot interaction. In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART)*, pages pp. 182–189, 2017. (Cited in page 118.)

[31] Matt Berlin, Jesse Gray, Andrea Lockerd Thomaz, and Cynthia Breazeal. Perspective taking: An organizing principle for learning in human-robot interaction. In *AAAI*, volume 2, pages 1444–1450, 2006. (Cited in page 34.)

[32] Cindy L. Bethel and Robin R. Murphy. Review of human studies methods in hri and recommendations. *International Journal of Social Robotics*, vol. 2(4):pp. 347–359, Dec 2010. (Cited in page 116.)

[33] Paul Bloom and Tim P German. Two reasons to abandon the false belief task as a test of theory of mind. *Cognition*, 77(1):B25–B31, 2000. (Cited in page 10.)

[34] Dan Bohus, Chit W Saw, and Eric Horvitz. Directions Robot: In-the-Wild Experiences and Lessons Learned. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, page 8, 2014. (Cited in page 133.)

[35] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, Inc., 2007. (Cited in pages 50 and 56.)

[36] Jeffrey M Bradshaw, Paul J Feltovich, and Matthew Johnson. Human–agent interaction. In *The handbook of human-machine interaction*, pages 283–300. CRC Press, 2017. (Cited in page 32.)

[37] Marcel Brass, Harold Bekkering, and Wolfgang Prinz. Movement observation affects movement execution in a simple response task. *Acta psychologica*, 106(1-2):3–22, 2001. (Cited in page 17.)

[38] Michael Bratman. Two faces of intention. *The Philosophical Review*, 93(3):375–405, 1984. (Cited in pages 14 and 15.)

[39] Michael Bratman et al. *Intention, plans, and practical reason*, volume 10. Harvard University Press Cambridge, MA, 1987. (Cited in page 40.)

[40] Michael E Bratman. Shared cooperative activity. *The philosophical review*, 101(2):327–341, 1992. (Cited in pages 18 and 21.)

[41] Michael E Bratman. Shared intention. *Ethics*, 104(1):97–113, 1993. (Cited in page 15.)

[42] Michael E Bratman. *Shared agency: A planning theory of acting together*. Oxford University Press, 2014. (Cited in page 16.)

[43] Michael E Bratman, David J Israel, and Martha E Pollack. Plans and resource-bounded practical reasoning. *Computational intelligence*, 4(3):349–355, 1988. (Cited in page 40.)

[44] Lars Braubach, Alexander Pokahr, and Winfried Lamersdorf. Jadex: A bdi-agent system combining middleware and reasoning. In Rainer Unland, Monique Calisti, and Matthias Klusch, editors, *Software Agent-Based Applications, Platforms and Development Kits*, pages 143–168, Basel, 2005. (Cited in page 50.)

[45] Jake Brawer, Olivier Mangin, Alessandro Roncone, Sarah Widder, and Brian Scassellati. Situated human–robot collaboration: predicting intent from grounded natural language. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 827–833. IEEE, 2018. (Cited in page 173.)

[46] Cynthia Breazeal. Regulation and entrainment in human—robot interaction. *The International Journal of Robotics Research*, 21(10-11):883–902, 2002. (Cited in pages 33 and 35.)

[47] Cynthia Breazeal. Emotion and sociable humanoid robots. *International journal of human-computer studies*, 59(1-2):119–155, 2003. (Cited in pages 33 and 35.)

[48] Cynthia Breazeal. Function meets style: insights from emotion theory applied to hri. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(2):187–194, 2004. (Cited in page 33.)

[49] Cynthia Breazeal et al. A motivational system for regulating human-robot interaction. In *AAAI-98 Proceedings*, pages 54–61, 1998. (Cited in page 33.)

[50] Susan E Brennan, Xin Chen, Christopher A Dickinson, Mark B Neider, and Gregory J Zelinsky. Coordinating cognition: The costs and benefits of shared gaze during collaborative search. *Cognition*, 106(3):1465–1477, 2008. (Cited in page 11.)

[51] Ingar Brinck. The role of intersubjectivity in the development of intentional communication. *The shared mind: Perspectives on intersubjectivity*, pages 115–140, 2008. (Cited in page 24.)

[52] Ingar Brinck and Christian Balkenius. Mutual recognition in human-robot interaction: A deflationary account. *Philosophy and Technology*, 1(1):53–70, 2018. (Cited in page 24.)

[53] Guilhem Buisan and Rachid Alami. A human-aware task planner explicitly reasoning about human and robot decision, action and reaction. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '21 Companion, page 544–548, New York, NY, USA, 2021. (Cited in pages 44, 65, and 94.)

[54] Guilhem Buisan, Guillaume Sarthou, Arthur Bit-Monnot, Aurélie Clodic, and Rachid Alami. Efficient, situated and ontology based referring expression generation for human-robot collaboration. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 349–356. IEEE, 2020. (Cited in pages 109 and 183.)

[55] Wolfram Burgard, Armin B Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The museum tour-guide robot rhino. In *Autonome Mobile Systeme 1998*, pages 245–254. Springer, 1999. (Cited in page 132.)

[56] George Butterworth and Nicholas Jarrett. What minds have in common is space: Spatial mechanisms serving joint visual attention in infancy. *British journal of developmental psychology*, 9(1):55–72, 1991. (Cited in page 20.)

[57] Michael D. Byrne and Susan Bovair. A working memory model of a common procedural error. *Cognitive Science*, 21(1):31–61, 1997. (Cited in page 153.)

[58] Luigia Camaioni, Paola Perucchini, Francesca Bellagamba, and Cristina Colonnesi. The role of declarative pointing in developing a theory of mind. *Infancy*, 5(3):291–308, 2004. (Cited in page 9.)

[59] Maxime Caniot, Vincent Bonnet, Maxime Busy, Thierry Labaye, Michel Besombes, Sebastien Courtois, and Edouard Lagrue. Adapted pepper. Technical report, SoftBank Robotics Europe, 2020. (Cited in page 154.)

[60] Malinda Carpenter. Just how joint is joint action in infancy? *Topics in Cognitive Science*, 1(2):380–392, 2009. (Cited in pages 11 and 12.)

[61] Malinda Carpenter and Kristin Liebal. Joint attention, communication, and knowing together in infancy. *Joint attention: New developments in psychology, philosophy of mind, and social neuroscience*, pages 159–181, 2011. (Cited in page 20.)

[62] Víctor Fernandez Castro, Aurélie Clodic, Rachid Alami, and Elisabeth Pacherie. Commitments in human-robot interaction. In *AI-HRI 2019 Proceedings*. 2019. (Cited in page 66.)

[63] Víctor Fernández Castro and Manuel Heras-Escribano. Social cognition: A normative approach. *Acta Analytica*, 35(1):75–100, 2020. (Cited in page 23.)

[64] Víctor Fernández Castro and Elisabeth Pacherie. Joint actions, commitments and the need to belong. *Synthese*, pages 1–30, 2020. (Cited in page 12.)

[65] Ravi Teja Chadalavada, Henrik Andreasson, Robert Krug, and Achim J Lilienthal. That's on my mind! robot to human intention communication through on-board projection on shared floor space. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2015. (Cited in pages 32 and 33.)

[66] Raphaël Chalmeau and Alain Gallo. La coopération chez les primates. *L'Année psychologique*, 95(1):119–130, 1995. (Cited in page 11.)

[67] Mai Lee Chang, Reymundo A Gutierrez, Priyanka Khante, Elaine Schaertl Short, and Andrea Lockerd Thomaz. Effects of integrated intent recognition and communication on human-robot collaboration. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3381–3386. IEEE, 2018. (Cited in page 34.)

[68] Marjorie H Charlop-Christy and Sabrina Daneshvar. Using video modeling to teach perspective taking to children with autism. *Journal of Positive Behavior Interventions*, 5(1):12–21, 2003. (Cited in page 9.)

[69] Yingfeng Chen, Feng Wu, Wei Shuai, and Xiaoping Chen. Robots serve humans in public places—kejia robot as a shopping assistant. *International Journal of Advanced Robotic Systems*, 14(3):1–20, 2017. (Cited in page 132.)

[70] Hui-Qing Chong, Ah-Hwee Tan, and Gee-Wah Ng. Integrated cognitive architectures: a survey. *Artificial Intelligence Review*, 28(2):103–130, 2007. (Cited in page 39.)

[71] Herbert H Clark. *Arenas of language use*. University of Chicago Press, 1992. (Cited in pages 21 and 23.)

[72] Herbert H Clark. *Using language*. Cambridge university press, 1996. (Cited in pages 11 and 21.)

[73] Herbert H Clark. Social actions, social commitments. In *Roots of human sociality*, pages 126–150. Routledge, 2006. (Cited in pages 12, 16, and 17.)

[74] Herbert H Clark and J Wade French. Telephone goodbyes. *Language in Society*, pages 1–19, 1981. (Cited in page 8.)

[75] Aurélie Clodic, Hung Cao, Samir Alili, Vincent Montreuil, Rachid Alami, and Raja Chatila. Shary: a supervision system adapted to human-robot interaction. In *Experimental robotics*, pages 229–238. Springer, 2009. (Cited in page 47.)

[76] Aurélie Clodic, Sara Fleury, Rachid Alami, Raja Chatila, Gérard Bailly, Ludovic Brethes, Maxime Cottret, Patrick Danes, Xavier Dollat, Frédéric Elisei, et al. Rackham: An interactive robot-guide. In *IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, pages 502–509. IEEE, 2006. (Cited in page 132.)

[77] Aurélie Clodic, Elisabeth Pacherie, Rachid Alami, and Raja Chatila. Key Elements for Human Robot Joint Action. In *Sociality and Normativity for RobotsPhilosophical Inquiries into Human-Robot Interactions*, Studies in the Philosophy of Sociality, pages 159–177. Springer, 2017. (Cited in page 13.)

[78] Philip R Cohen and Hector J Levesque. Intention is choice with commitment. *Artificial intelligence*, 42(2-3):213–261, 1990. (Cited in pages 15 and 40.)

[79] Philip R Cohen and Hector J Levesque. Teamwork. *Nous*, 25(4):487–512, 1991. (Cited in pages 11, 15, 16, 18, 21, and 33.)

[80] Richard Cooper and Tim Shallice. Contention scheduling and the control of routine activities. *Cognitive neuropsychology*, 17(4):297–338, 2000. (Cited in page 18.)

[81] Michael D Coovert, Tiffany Lee, Ivan Shindev, and Yu Sun. Spatial augmented reality as a method for a mobile robot to communicate intended movement. *Computers in Human Behavior*, 34:241–248, 2014. (Cited in pages 32 and 33.)

[82] Arianna Curioni, Gunther Knoblich, and Natalie Sebanz. *Joint Action in Humans: A Model for Human-Robot Interactions*, pages 1–19. Springer Netherlands, Dordrecht, 2017. (Cited in pages 12, 133, and 134.)

[83] Kerstin Dautenhahn, Bernard Ogden, and Tom Quick. From embodied to socially embedded agents–implications for interaction-aware robots. *Cognitive Systems Research*, 3(3):397–428, 2002. (Cited in page 30.)

[84] Jenny L Davis and Tony P Love. Self-in-self, mind-in-mind, heart-in-heart: The future of role-taking, perspective taking, and empathy. In *Advances in group processes*. Emerald Publishing Limited, 2017. (Cited in page 9.)

[85] Daniel C. Dennett. *Brainstorms: Philosophical Essays on Mind and Psychology*. Bradford Books, 1978. (Cited in page 10.)

[86] Sandra Devin. *Decisional issues during human-robot joint action*. PhD thesis, 2017. Thèse de doctorat dirigée par Alami, Rachid et Ghallab, Malik Intelligence Artificielle Toulouse, INPT 2017. (Cited in pages 48 and 107.)

[87] Sandra Devin and Rachid Alami. An implemented theory of mind to improve human-robot shared plans execution. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 319–326. IEEE, 2016. (Cited in pages 34, 48, 67, and 92.)

[88] Sandra Devin, Aurélie Clodic, and Rachid Alami. About decisions during human-robot shared plan achievement: Who should act and how? In *International Conference on Social Robotics*, pages 453–463. Springer, 2017. (Cited in pages 48, 71, 92, and 101.)

[89] Mark d'Inverno, David Kinny, Michael Luck, and Michael Wooldridge. A formal specification of dmars. In Munindar P. Singh, Anand Rao, and Michael J. Wooldridge, editors, *Intelligent Agents IV Agent Theories, Architectures, and Languages*, pages 155–176. Springer Berlin Heidelberg, 1998. (Cited in page 50.)

[90] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308. IEEE, 2013. (Cited in page 33.)

[91] Iroise Dumontheil, Ian A Apperly, and Sarah-Jayne Blakemore. Online usage of theory of mind continues to develop in late adolescence. *Developmental science*, 13(2):331–338, 2010. (Cited in page 175.)

[92] Nathan J Emery. The eyes have it: the neuroethology, function and evolution of social gaze. *Neuroscience & biobehavioral reviews*, 24(6):581–604, 2000. (Cited in pages 21 and 22.)

[93] J. Fan, D. Bian, Z. Zheng, L. Beuscher, P. A. Newhouse, L. C. Mion, and N. Sarkar. A robotic coach architecture for elder care (rocare) based on multi-user engagement models. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25(8):pp. 1153–1163, 2017. (Cited in pages 121 and 122.)

[94] Víctor Fernández Castro, Amandine Mayima, Kathleen Belhassein, and Aurélie Clodic. The role of commitments in socially appropriate robotics. submitted. (Cited in page 10.)

[95] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596. IEEE, 2005. (Cited in page 181.)

[96] Anika Fiebich and Shaun Gallagher. Joint attention in joint action. *Philosophical Psychology*, 26(4):571–587, 2013. (Cited in page 11.)

[97] John H Flavell, Patricia T Botkin, Charles L Fry, John W Wright, and Paul E Jarvis. The development of role-taking and communication skills in children. 1968. (Cited in page 9.)

[98] Mary Ellen Foster, Bart Craenen, Amol Deshmukh, Oliver Lemon, Emanuele Bastianelli, Christian Dondrup, and et al. Mummer: Socially intelligent human-robot interaction in public spaces. In *AAAI 2019 Fall Symposium Series*, Arlington, United States, November 2019. (Cited in page 157.)

[99] Mary Ellen Foster, Bart Craenen, Amol Deshmukh, Oliver Lemon, Emanuele Bastianelli, Christian Dondrup, Ioannis Papaioannou, Andrea Vanzo, Jean-Marc Odobez, Olivier Canévet, Yuanzhouhan Cao, Weipeng He, Angel Martínez-González, Petr Motlicek, Rémy Siegfried, Rachid Alami, Kathleen Belhassein, Guilhem Buisan, Aurélie Clodic, Amandine Mayima, Yoan Sallami, Guillaume Sarthou, Phani-Teja Singamaneni, Jules Waldhart, Alexandre Mazel, Maxime Caniot, Marketta Niemelä, Päivi Heikkilä, Hanna Lammi, and Antti Tammela. Mummer: Socially intelligent human-robot interaction in public spaces. In *Artificial Intelligence for Human-Robot Interaction Symposium (AI-HRI)*, Arlington, VA, United States, 2019. AAAI Fall Symposium Series 2019. (Cited in page 139.)

[100] Andre Gaschler, Kerstin Huth, Manuel Giuliani, Ingmar Kessler, Jan de Ruiter, and Alois Knoll. Modelling state of interaction from head poses for social human-robot interaction. In *Proceedings of the Gaze in Human-Robot Interaction Workshop held at the 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2012)*, 2012. (Cited in pages 28 and 29.)

[101] M Georgeff and A Rao. Modeling rational agents within a bdi-architecture. In *Proc. 2nd Int. Conf. on Knowledge Representation and Reasoning (KR'91). Morgan Kaufmann*, pages 473–484. of, 1991. (Cited in page 40.)

[102] Michael Georgeff and Felix Ingrand. Decision-making in an embedded reasoning system. In *International Joint Conference on Artificial Intelligence*, 1989. (Cited in page 40.)

[103] Malik Ghallab, Craig Knoblock, David Wilkins, Anthony Barrett, Dave Christianson, Marc Friedman, and et al. *PDDL - The Planning Domain Definition Language*, 08 1998. (Cited in page 67.)

[104] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016. (Cited in page 65.)

[105] James J Gibson. The theory of affordances. In *The Ecological Approach to Visual Perception*, pages 127–137. Houghton Mifflin, 1979. (Cited in page 19.)

[106] Margaret Gilbert. *On social facts*. Routledge, 1989. (Cited in page 15.)

[107] Margaret Gilbert. Shared intention and personal intentions. *Philosophical studies*, 144(1):167–187, 2009. (Cited in pages 16 and 17.)

[108] Margaret Gilbert. *Joint commitment: How we make the social world*. Oxford University Press, 2013. (Cited in page 17.)

[109] Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid Simmons, Kevin Snipes, Alan C Schultz, et al. Designing robots for long-term social interaction. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1338–1343. IEEE, 2005. (Cited in page 29.)

[110] Marion Godman. Why we do things together: The social motivation for joint action. *Philosophical Psychology*, 26(4):588–603, 2013. (Cited in page 11.)

[111] Erving Goffman. *Interaction ritual: Essays on face-to-face interaction*. Aldine, 1967. (Cited in page 6.)

[112] Erving Goffman. The interaction order: American sociological association, 1982 presidential address. *American sociological review*, 48(1):1–17, 1983. (Cited in page 30.)

[113] O Can Görür, Benjamin Rosman, Fikret Sivrikaya, and Sahin Albayrak. Social cobots: Anticipatory decision-making for collaborative robots incorporating unexpected human behaviors. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 398–406, 2018. (Cited in page 48.)

[114] Maria Gräfenhain, Malinda Carpenter, and Michael Tomasello. Three-year-olds' understanding of the consequences of joint commitments. *PLoS One*, 8(9):e73039, 2013. (Cited in page 11.)

[115] Herbert P Grice. Logic and conversation. In *Speech acts*, pages 41–58. Brill, 1975. (Cited in page 190.)

[116] Paul Grice. *Studies in the Way of Words*. Harvard University Press, 1989. (Cited in page 23.)

[117] H-M Gross, H Boehme, Ch Schroeter, Steffen Müller, Alexander König, Erik Einhorn, Ch Martin, Matthias Merten, and Andreas Bley. Toomas: interactive shopping guide robots in everyday use-final implementation and experiences from long-term field trials. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2005–2012. IEEE, 2009. (Cited in page 132.)

[118] Barbara J Grosz and Sarit Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996. (Cited in page 19.)

[119] Orhan Görür, Benjamin Rosman, Guy Hoffman, and Sahin Albayrak. Toward integrating theory of mind into adaptive decision- making of social robots to understand human intention. In *Workshop on The Role of Intentions in Human-Robot Interaction in 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI'17)*, 03 2017. (Cited in page 48.)

[120] Adriana Hamacher, Nadia Bianchi-Berthouze, Anthony G Pipe, and Kerstin Eder. Believing in bert: Using expressive communication to enhance trust and counteract operational error in physical human-robot interaction. In *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)*, pages 493–500. IEEE, 2016. (Cited in page 35.)

[121] Francesca GE Happé. An advanced test of theory of mind: Understanding of story characters' thoughts and feelings by able autistic, mentally handicapped, and normal children and adults. *Journal of autism and Developmental disorders*, 24(2):129–154, 1994. (Cited in page 175.)

[122] Nick Hawes, Michael Zillich, and Jeremy Wyatt. Balt & cast: Middleware for cognitive robotics. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 998–1003. IEEE, 2007. (Cited in page 180.)

[123] Meghan L Healey and Murray Grossman. Cognitive and affective perspective-taking: evidence for shared and dissociable anatomical substrates. *Frontiers in neurology*, 9:491, 2018. (Cited in page 10.)

[124] Raphaela Heesen, Emilie Genty, Federico Rossano, Klaus Zuberbühler, and Adrian Bangerter. Social play as joint action: A framework to study the evolution of shared intentionality as an interactional achievement. *Learning & behavior*, 45(4):390–405, 2017. (Cited in page 12.)

[125] Päivi Heikkilä, Hanna Lammi, and Kathleen Belhassein. Where can i find a pharmacy? -human-driven design of a service robot's guidance behaviour. In *Workshop on Public Space Human-Robot Interaction (PubRob) as part of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 1–2, 2018. (Cited in page 135.)

[126] Päivi Heikkilä, Hanna Lammi, Marketta Niemelä, Kathleen Belhassein, Guillaume Sarthou, Antti Tammela, Aurélie Clodic, and Rachid Alami. Should a robot guide like a human? a qualitative four-phase study of a shopping mall robot. In *International Conference on Social Robotics (ICSR)*, pages 548–557. Springer, 2019. (Cited in page 135.)

[127] Laura M Hiatt, Anthony M Harrison, and J Gregory Trafton. Accommodating human variability in human-robot teams through theory of mind. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. (Cited in page 34.)

[128] Laura M. Hiatt, Cody Narber, Esube Bekele, Sangeet S. Khemlani, and J. Gregory Trafton. Human modeling for human-robot collaboration. *International Journal of Robotics Research*, vol. 36(5-7):pp. 580–596, 2017. (Cited in page 67.)

[129] Laura M Hiatt and J Gregory Trafton. A cognitive model of theory of mind. In *Proceedings of the 10th international conference on cognitive modeling*, pages 91–96. Citeseer, 2010. (Cited in page 34.)

[130] G. Hoffman. Evaluating fluency in human–robot collaboration. *IEEE Transactions on Human-Machine Systems*, 49(3):pp. 209–218, June 2019. (Cited in pages 117 and 169.)

[131] G. Hoffman and C. Breazeal. Cost-based anticipatory action selection for human–robot fluency. *IEEE Transactions on Robotics*, vol. 23(5):pp. 952–961, 2007. (Cited in page 115.)

[132] Shanee Honig and Tal Oron-Gilad. Understanding and resolving failures in human-robot interaction: Literature review and model development. *Frontiers in psychology*, 9:861, 2018. (Cited in pages 34 and 35.)

[133] Andrew Howes and Richard M. Young. The role of cognitive architecture in modeling the user: Soar's learning mechanism. *Human–Computer Interaction*, 12(4):311–343, 1997. (Cited in page 39.)

[134] Chien-Ming Huang and Andrea L Thomaz. Joint attention in human-robot interaction. In *2010 AAAI Fall Symposium Series*, 2010. (Cited in page 33.)

[135] Marcus J. Huber. Jam: A bdi-theoretic mobile agent architecture. In *Proceedings of the Third Annual Conference on Autonomous Agents*, AGENTS '99, page 236–243, 1999. (Cited in page 50.)

[136] Catherine A. Hynes, Abigail A. Baird, and Scott T. Grafton. Differential role of the orbital frontal lobe in emotional versus cognitive perspective-taking. *Neuropsychologia*, 44(3):374–383, 2006. (Cited in page 10.)

[137] M. Imai, T. Ono, and H. Ishiguro. Physical relation and expression: joint attention for human-robot interaction. *IEEE Transactions on Industrial Electronics*, 50(4):636–643, 2003. (Cited in page 32.)

[138] Félix Ingrand and Malik Ghallab. Deliberation for autonomous robots: A survey. *Artificial Intelligence*, vol. 247:pp. 10–44, June 2017. (Cited in page 65.)

[139] F.F. Ingrand, R. Chatila, R. Alami, and F. Robert. Prs: a high level supervision and control language for autonomous mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 43–49 vol.1, 1996. (Cited in page 50.)

[140] Luca Iocchi, Laurent Jeanpierre, Maria Teresa Lazaro, and Abdel-Illah Mouaddib. A practical framework for robust decision-theoretic planning and execution for service robots. In *Twenty-Sixth International Conference on Automated Planning and Scheduling*, 2016. (Cited in page 48.)

[141] Luca Iocchi, Maria Teresa Lázaro, Laurent Jeanpierre, and Abdel-Illah Mouaddib. Personalized Short-Term Multi-modal Interaction for Social Robots Assisting Users in Shopping Malls. In Adriana Tapus, Elisabeth André, Jean-Claude Martin, François Ferland, and Mehdi Ammi, editors, *Social Robotics*, volume 9388, pages 264–274. Springer International Publishing, Cham, 2015. (Cited in pages 28 and 133.)

[142] K. Itoh, H. Miwa, Y. Nukariya, M. Zecca, H. Takanobu, S. Roccella, and et al. Development of a bioinstrumentation system in the interaction between a human and a robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages pp. 2620–2625, Beijing, China, Oct 2006. (Cited in page 117.)

[143] Lorenzo Jamone, Emre Ugur, Angelo Cangelosi, Luciano Fadiga, Alexandre Bernardino, Justus Piater, and José Santos-Victor. Affordances in psychology, neuroscience, and robotics: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(1):4–25, 2016. (Cited in page 19.)

[144] Matthew Johnson and Yiannis Demiris. Perceptual perspective taking and action recognition. *International Journal of Advanced Robotic Systems*, 2(4):32, 2005. (Cited in page 34.)

[145] Peter H Kahn, Nathan G Freier, Takayuki Kanda, Hiroshi Ishiguro, Jolina H Ruckert, Rachel L Severson, and Shaun K Kane. Design patterns for sociality in human-robot interaction. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 97–104, 2008. (Cited in page 30.)

[146] Peter H Kahn, Brian T Gill, Aimee L Reichert, Takayuki Kanda, Hiroshi Ishiguro, and Jolina H Ruckert. Validating interaction patterns in hri. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 183–184. IEEE, 2010. (Cited in page 30.)

[147] Takayuki Kanda, Rumi Sato, Naoki Saiwaki, and Hiroshi Ishiguro. A two-month field trial in an elementary school for long-term human–robot interaction. *IEEE Transactions on Robotics*, 23(5):962–971, 2007. (Cited in page 28.)

[148] Takayuki Kanda, Masahiro Shiomi, Zenta Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. An affective guide robot in a shopping mall. In *ACM/IEEE international Conference on Human-Robot interaction (HRI)*, pages 173–180, 2009. (Cited in pages 132 and 151.)

[149] Takayuki Kanda, Masahiro Shiomi, Zenta Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. A communication robot in a shopping mall. *IEEE Transactions on Robotics*, 26(5):897–913, 2010. (Cited in page 132.)

[150] Frederic Kaplan and Verena V Hafner. The challenges of joint attention. *Interaction Studies*, 7(2):135–169, 2006. (Cited in pages 19, 20, and 21.)

[151] Erez Karpas, Steven J Levine, Peng Yu, and Brian C Williams. Robust execution of plans for human-robot teams. In *Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015. (Cited in page 47.)

[152] Zerrin Kasap and Nadia Magnenat-Thalmann. Building long-term relationships with virtual and robotic characters: the role of remembering. *The Visual Computer*, 28(1):87–97, 2012. (Cited in page 29.)

[153] Adam Kendon. *Conducting interaction: Patterns of behavior in focused encounters*, volume 7. CUP Archive, 1990. (Cited in pages 7, 30, and 146.)

[154] Boaz Keysar. The illusory transparency of intention: Linguistic perspective taking in text. *Cognitive psychology*, 26(2):165–208, 1994. (Cited in page 175.)

[155] Boaz Keysar and Dale J Barr. Self-anchoring in conversation: Why language users do not do what they'should'. *Heuristics and biases: The psychology of intuitive judgment*, 2002. (Cited in page 175.)

[156] Boaz Keysar, Dale J Barr, Jennifer A Balin, and Jason S Brauner. Taking perspective in conversation: The role of mutual knowledge in comprehension. *Psychological Science*, 11(1):32–38, 2000. (Cited in page 174.)

[157] Boaz Keysar, Dale J Barr, and William S Horton. The egocentric basis of language use: Insights from a processing approach. *Current directions in psychological science*, 7(2):46–49, 1998. (Cited in page 175.)

[158] Boaz Keysar, Shuhong Lin, and Dale J Barr. Limits on theory of mind use in adults. *Cognition*, 89(1):25–41, 2003. (Cited in page 175.)

[159] Harmish Khambhaita and Rachid Alami. Viewing robot navigation in human environment as a cooperative activity. In Nancy M. Amato, Greg Hager, Shawna Thomas, and Miguel Torres-Torriti, editors, *Robotics Research*, pages pp. 285–300. Springer International Publishing, 2020. (Cited in page 123.)

[160] Cory D Kidd and Cynthia Breazeal. Robots at home: Understanding long-term human-robot interaction. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3230–3235. IEEE, 2008. (Cited in page 29.)

[161] Hyoung-Rock Kim and Dong-Soo Kwon. Computational model of emotion generation for human–robot interaction based on the cognitive appraisal theory. *Journal of Intelligent & Robotic Systems*, 60(2):263–283, 2010. (Cited in page 33.)

[162] Gary Klein, Paul J. Feltovich, Jeffrey M. Bradshaw, and David D. Woods. *Common Ground and Coordination in Joint Activity*, chapter 6, pages 139–184. John Wiley & Sons, Ltd, 2005. (Cited in page 11.)

[163] Mark L Knapp, Roderick P Hart, Gustav W Friedrich, and Gary M Shulman. The rhetoric of goodbye: Verbal and nonverbal correlates of human leave-taking. *Communications Monographs*, 40(3):182–198, 1973. (Cited in page 8.)

[164] Ross A Knepper, Stefanie Tellex, Adrian Li, Nicholas Roy, and Daniela Rus. Recovering from failure by asking for help. *Autonomous Robots*, 39(3):347–362, 2015. (Cited in page 36.)

[165] Günther Knoblich, Stephen Butterfill, and Natalie Sebanz. Chapter three - psychological research on joint action: Theory and data. In Brian H. Ross, editor, *Advances in Research and Theory*, volume 54 of *Psychology of Learning and Motivation*, pages 59–101. Academic Press, 2011. (Cited in pages 17, 19, 22, and 134.)

[166] Harumi Kobayashi, Tetsuya Yasuda, Hiroshi Igarashi, and Satoshi Suzuki. Language use in joint action: The means of referring expressions. *International Journal of Social Robotics*, pages 1–9, 2018. (Cited in page 11.)

[167] Stefan Kopp, Paul A. Tepper, Kimberley Ferriman, Kristina Striegnitz, and Justine Cassell. *Trading Spaces: How Humans and Humanoids Use Speech and Gesture to Give Directions*, chapter 8, pages 133–160. John Wiley & Sons, Ltd, 2007. (Cited in page 132.)

[168] Iuliia Kotseruba and John K Tsotsos. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1):17–94, 2020. (Cited in page 39.)

[169] Robert M Krauss and Susan R Fussell. Perspective-taking in communication: Representations of others' knowledge in reference. *Social cognition*, 9(1):2–24, 1991. (Cited in page 10.)

[170] Robert M Krauss and Sam Glucksberg. Social and nonsocial speech. *Scientific American*, 236(2):100–105, 1977. (Cited in page 174.)

[171] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-Aware Robot Navigation: A Survey. *Robotics and Autonomous Systems*, vol. 61(12):pp. 1726–1743, December 2013. (Cited in page 115.)

[172] D. Kulic and E. A. Croft. Affective state estimation for human–robot interaction. *IEEE Transactions on Robotics*, vol. 23(5):pp. 991–1000, 2007. (Cited in page 117.)

[173] Dana Kulić and Elizabeth A. Croft. Estimating intent for human-robot interaction. In *IEEE International Conference on Advanced Robotics*, pages pp. 810–815, 2003. (Cited in page 117.)

[174] I-Han Kuo. *Designing Human-Robot Interaction for Service Applications*. PhD thesis, ResearchSpace@ Auckland, 2012. (Cited in pages 30 and 32.)

[175] Minae Kwon, Sandy H. Huang, and Anca D. Dragan. Expressing robot incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, page 87–95, New York, NY, USA, 2018. (Cited in page 35.)

[176] Raphaël Lallement, Lavindra De Silva, and Rachid Alami. HATP: An HTN Planner for Robotics. In *2nd ICAPS Workshop on Planning and Robotics*, Portsmouth, United States, June 2014. (Cited in pages 44 and 65.)

[177] Philippe Lamarre and Yoav Shoham. Knowledge, certainty, belief, and conditionalisation (abbreviated version). In *Principles of knowledge representation and reasoning*, pages 415–424. Elsevier, 1994. (Cited in page 40.)

[178] David A. Leavens, William D. Hopkins, and Roger K. Thomas. Referential communication by chimpanzees (pan troglodytes). *Journal of Comparative Psychology*, 118(1):48–57, 2004. (Cited in page 24.)

[179] John O. Ledyard. Public Goods: A Survey of Experimental Research. Public Economics 9405003, University Library of Munich, Germany, May 1994. (Cited in page 16.)

[180] Min Kyung Lee, Jodi Forlizzi, Sara Kiesler, Paul Rybski, John Antanitis, and Sarun Savetsila. Personalization in hri: A longitudinal field experiment. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 319–326. IEEE, 2012. (Cited in page 30.)

[181] Min Kyung Lee, Sara Kiesler, Jodi Forlizzi, Siddhartha Srinivasa, and Paul Rybski. Gracefully mitigating breakdowns in robotic services. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 203–210. IEEE, 2010. (Cited in page 36.)

[182] Iolanda Leite, Carlos Martinho, and Ana Paiva. Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308, 2013. (Cited in page 28.)

[183] S. Lemaignan, F. Garcia, A. Jacq, and P. Dillenbourg. From real-time attention assessment to "with-me-ness" in human-robot interaction. In *11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages pp. 157–164, 2016. (Cited in page 122.)

[184] Séverin Lemaignan, Yoan Sallami, Christopher Wallhridge, Aurélic Clodic, Tony Belpaeme, and Rachid Alami. Underworlds: cascading situation assessment for robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7750–7757. IEEE, 2018. (Cited in pages 42, 143, and 181.)

[185] Séverin Lemaignan, Mathieu Warnier, E Akin Sisbot, Aurélie Clodic, and Rachid Alami. Artificial cognition for social human–robot interaction: An implementation. *Artificial Intelligence*, 247:45–69, 2017. (Cited in pages 40, 41, and 115.)

[186] Alan M Leslie. Spatiotemporal continuity and the perception of causality in infants. *Perception*, 13(3):287–305, 1984. (Cited in page 34.)

[187] David Lewis. *Convention: A philosophical study*. Wiley-Blackwell, 1969. (Cited in page 21.)

[188] Shuyin Li, Britta Wrede, and Gerhard Sagerer. A computational model of multi-modal grounding for human robot interaction. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 153–160, 2006. (Cited in page 35.)

[189] Shuhong Lin, Boaz Keysar, and Nicholas Epley. Reflexively mindblind: Using theory of mind to interpret behavior requires effortful attention. *Journal of Experimental Social Psychology*, 46(3):551–556, 2010. (Cited in page 175.)

[190] Jim Mainprice, Mamoun Gharbi, Thierry Siméon, and Rachid Alami. Sharing effort in planning human-robot handover tasks. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 764–770. IEEE, 2012. (Cited in page 43.)

[191] Célia Martinie, Philippe Palanque, and Marco Winckler. Structuring and composition mechanisms to address scalability issues in task models. In *IFIP*

Conference on Human-Computer Interaction, pages 589–609. Springer, 2011. (Cited in page 79.)

[192] Robert S Marvin, Mark T Greenberg, and Daniel G Mossler. The early development of conceptual perspective taking: Distinguishing among multiple perspectives. *Child Development*, pages 511–514, 1976. (Cited in page 9.)

[193] Takahiro Matsumoto, Satoru Satake, Takayuki Kanda, Michita Imai, and Norihiro Hagita. Do you remember that shop? computational model of spatial memory for shopping companion robots. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 447–454, 2012. (Cited in pages 132 and 139.)

[194] Alyxander David May, Christian Dondrup, and Marc Hanheide. Show me your moves! conveying navigation intention of a mobile robot to humans. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2015. (Cited in page 33.)

[195] A. Mayima, A. Clodic, and R. Alami. Towards robots able to measure in real-time the quality of interaction. 2021. (Cited in pages 115 and 161.)

[196] David McNeill. Gesture, gaze, and ground. In *International workshop on machine learning for multimodal interaction*, pages 1–14. Springer, 2005. (Cited in page 146.)

[197] John Michael and Elisabeth Pacherie. On commitments and other uncertainty reduction tools in joint action. *Journal of Social Ontology*, 1(1):89–120, 2015. (Cited in pages 16, 22, and 23.)

[198] John Michael and Alessandro Salice. The sense of commitment in human–robot interaction. *International Journal of Social Robotics*, vol. 9(5):pp. 755–763, Nov 2017. (Cited in pages 16 and 67.)

[199] John Michael, Natalie Sebanz, and Günther Knoblich. The sense of commitment: A minimal approach. *Frontiers in Psychology*, vol. 6:1968, 2016. (Cited in pages 17 and 121.)

[200] Grégoire Milliez, Raphaël Lallement, Michelangelo Fiore, and Rachid Alami. Using human knowledge awareness to adapt collaborative plan generation, explanation and monitoring. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 43–50. IEEE, 2016. (Cited in page 34.)

[201] Grégoire Milliez, Matthieu Warnier, Aurélie Clodic, and Rachid Alami. A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*,

pages pp. 1103–1109, Edinburgh, United Kingdom, August 2014. (Cited in pages 34, 42, 67, 173, and 179.)

[202] Stephen Monsell. Task switching. *Trends in cognitive sciences*, 7(3):134–140, 2003. (Cited in page 18.)

[203] Cecilia G Morales, Elizabeth J Carter, Xiang Zhi Tan, and Aaron Steinfeld. Interaction needs and opportunities for failing robots. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, pages 659–670, 2019. (Cited in page 35.)

[204] Y. Morales, Satoru Satake, Takayuki Kanda, and Norihiro Hagita. Modeling environments from a route perspective. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 441–448, 2011. (Cited in page 132.)

[205] D. Morley and K. Myers. The spark agent framework. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, pages 714–721, 2004. (Cited in page 50.)

[206] Clément Moulin-Frier, Tobias Fischer, Maxime Petit, Grégoire Pointeau, Jordi-Ysard Puigbo, Ugo Pattacini, Sock Ching Low, Daniel Camilleri, Phuong Nguyen, Matej Hoffmann, et al. Dac-h3: a proactive robot cognitive architecture to acquire and express knowledge about the world and the self. *IEEE Transactions on Cognitive and Developmental Systems*, 10(4):1005–1022, 2017. (Cited in page 40.)

[207] J. R. Movellan, F. Tanaka, I. R. Fasel, C. Taylor, P. Ruvolo, and M. Eckhardt. The rubi project: A progress report. In *2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages pp. 333–339, 2007. (Cited in page 170.)

[208] Bilge Mutlu, Allison Terrell, and Chien-Ming Huang. Coordination mechanisms in human-robot collaboration. In *Proceedings of the Workshop on Collaborative Manipulation, 8th ACM/IEEE International Conference on Human-Robot Interaction*, pages 1–6, 2013. (Cited in page 36.)

[209] Allen Newell. *Unified theories of cognition*. Harvard University Press, 1994. (Cited in page 27.)

[210] Kai Nickel and Rainer Stiefelhagen. Visual recognition of pointing gestures for human–robot interaction. *Image and vision computing*, 25(12):1875–1884, 2007. (Cited in page 33.)

[211] Donald A. Norman. Categorization of action slips. *Psychological Review*, 88(1):1–15, 1981. (Cited in page 25.)

[212] Bernard Ogden, Kerstin Dautenhahn, and Penny Stribling. Interactional structure applied to the identification and generation of visual interactive behavior: Robots that (usually) follow the rules. In *International Gesture Workshop*, pages 254–268. Springer, 2001. (Cited in page 30.)

[213] Yusuke Okuno, Takayuki Kanda, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. Providing route directions: design of robot's utterance, gesture, and timing. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 53–60. IEEE, 2009. (Cited in pages 132 and 139.)

[214] Dan R. Olsen and Michael A. Goodrich. Metrics for evaluating human-robot interaction. In *PERMIS*, Gaithersburg, United States, 2003. (Cited in page 122.)

[215] Richard Osborne. *An ecological approach to educational technology: affordance as a design tool for aligning pedagogy and technology.* PhD thesis, University of Exeter, 2014. (Cited in page 19.)

[216] Elisabeth Pacherie. The phenomenology of action: A conceptual framework. *Cognition*, 107(1):179–217, 2008. (Cited in page 13.)

[217] Elisabeth Pacherie. The Phenomenology of Joint Action: Self-Agency vs. Joint-Agency. In Axel Seemann, editor, *Joint Attention: New Developments*, pages 343–389. MIT Press, 2012. (Cited in pages 11, 13, 17, 18, 19, 22, and 42.)

[218] Elisabeth Pacherie. Intentional joint agency: shared intention lite. *Synthese*, 190(10):1817–1839, 2013. (Cited in pages 9 and 12.)

[219] Julia Peltason and Britta Wrede. Pamini: A framework for assembling mixed-initiative human-robot interaction from generic interaction patterns. In *Proceedings of the SIGDIAL 2010 Conference*, pages 229–232, 2010. (Cited in page 32.)

[220] Josef Perner and Heinz Wimmer. "john thinks that mary thinks that..." attribution of second-order beliefs by 5-to 10-year-old children. *Journal of experimental child psychology*, 39(3):437–471, 1985. (Cited in page 9.)

[221] Ronald PA Petrick, Mary Ellen Foster, and Amy Isard. Social state recognition and knowledge-level planning for human-robot interaction in a bartender domain. In *AAAI Workshop on Grounding Language for Physical Systems, Toronto, ON, Canada, July*, 2012. (Cited in page 173.)

[222] Gregoire Pointeau and Peter Ford Dominey. The role of autobiographical memory in the development of a robot self. *Frontiers in neurorobotics*, 11:27, 2017. (Cited in page 34.)

[223] Daniel J Povinelli and Jennifer Vonk. We don't need a microscope to explore the chimpanzee's mind. *Mind & Language*, 19(1):1–28, 2004. (Cited in page 8.)

[224] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526, 1978. (Cited in page 8.)

[225] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009. (Cited in pages 41 and 57.)

[226] Verónica C Ramenzoni, Michael A Riley, Kevin Shockley, and Tehran Davis. Short article: Carrying the height of the world on your ankles: Encumbering observers reduces estimates of how high an actor can jump. *Quarterly Journal of Experimental Psychology*, 61(10):1487–1495, 2008. (Cited in page 17.)

[227] Anand S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In *Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Agents Breaking Away*, pages 42–55, Berlin, Heidelberg, 1996. (Cited in page 50.)

[228] Anand S Rao and Michael P Georgeff. Bdi agents: From theory to practice. In *ICMAS*, volume 95, pages 312–319, 1995. (Cited in page 40.)

[229] Jens Rasmussen. Human errors. a taxonomy for describing human malfunction in industrial installations. *Journal of Occupational Accidents*, 4(2):311–333, 1982. (Cited in page 25.)

[230] James Reason. *Human error*. Cambridge university press, 1990. (Cited in pages 25 and 26.)

[231] Mauricio Reyes, Ivan Meza, and Luis A Pineda. The positive effect of negative feedback in hri using a facial expression robot. In *International Workshop on Cultural Robotics*, pages 44–54. Springer, 2015. (Cited in page 35.)

[232] Michael J Richardson, Kerry L Marsh, and Reuben M Baron. Judging and actualizing intrapersonal and interpersonal affordances. *Journal of experimental psychology: Human Perception and Performance*, 33(4):845, 2007. (Cited in page 19.)

[233] Giacomo Rizzolatti and Laila Craighero. The mirror-neuron system. *Annu. Rev. Neurosci.*, 27:169–192, 2004. (Cited in page 19.)

[234] Jeffrey D. Robinson. *The handbook of conversation analysis*, volume 121, chapter Overall Structural Organization. John Wiley & Sons, 2012. (Cited in pages 7, 8, and 65.)

[235] Raquel Ros, E Akin Sisbot, Rachid Alami, Jasmin Steinwender, Katharina Hamann, and Felix Warneken. Solving ambiguities with perspective taking. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 181–182. IEEE, 2010. (Cited in page 34.)

[236] Abraham Sesshu Roth. Shared agency and contralateral commitments. *The Philosophical Review*, 113(3):359–410, 2004. (Cited in pages 16 and 17.)

[237] Paula Rubio-Fernández. The director task: A test of theory-of-mind use or selective attention? *Psychonomic bulletin & review*, 24(4):1121–1128, 2017. (Cited in page 175.)

[238] Rudolph J Rummel. Understanding conflict and war: vol. 2: the conflict helix. *Beverly Hills: Sage*, 1976. (Cited in page 6.)

[239] Harvey Sacks. *Lectures on Conversation*. Wiley-Blackwell, Oxford, volumes i and ii edition edition, January 1995. (Cited in page 7.)

[240] Ricardo Sanchez-Matilla, Konstantinos Chatzilygeroudis, Apostolos Modas, Nuno Ferreira Duarte, Alessio Xompero, Pascal Frossard, Aude Billard, and Andrea Cavallaro. Benchmark for human-to-robot handovers of unseen containers with unknown filling. *IEEE Robotics and Automation Letters*, 5(2):1642–1649, 2020. (Cited in page 170.)

[241] Valerio Sanelli, Michael Cashmore, Daniele Magazzeni, and Luca Iocchi. Short-term human-robot interaction through conditional planning and execution. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 27, 2017. (Cited in page 28.)

[242] Idalmis Santiesteban, Sarah White, Jennifer Cook, Sam J Gilbert, Cecilia Heyes, and Geoffrey Bird. Training social cognition: from imitation to theory of mind. *Cognition*, 122(2):228–235, 2012. (Cited in page 175.)

[243] Guillaume Sarthou, Mayima Amandine, Buisan Guilhem, Belhassein Kathleen, and Aurélie Clodic. The director task: a psychology-inspired task to assess cognitive and interactive robot architectures. In *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–8. IEEE, 2021. (Cited in page 172.)

[244] Guillaume Sarthou, Aurélie Clodic, and Rachid Alami. Ontologenius: A long-term semantic memory for robotic agents. In *28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–8. IEEE, 2019. (Cited in pages 42 and 140.)

[245] Guillaume Sarthou, Aurélie Clodic, and Rachid Alami. Semantic spatial representation: a unique representation of an environment based on an ontology for robotic applications. In *Proceedings of the Combined Workshop on Spatial Language Understanding (SpLU) and Grounded Communication for Robotics (RoboNLP)*, pages pp.50–60, Minneapolis, United States, 2019. (Cited in pages 142 and 144.)

[246] Satoru Satake, Kotaro Hayashi, Keita Nakatani, and Takayuki Kanda. Field trial of an information-providing robot in a shopping mall. In *IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pages 1832–1839. IEEE, 2015. (Cited in pages 139 and 140.)

[247] Satoru Satake, Keita Nakatani, Kotaro Hayashi, Takyuki Kanda, and Michita Imai. What should we know to develop an information robot? *PeerJ Computer Science*, 1:8, 2015. (Cited in pages 84, 133, 139, and 172.)

[248] Allison Sauppé and Bilge Mutlu. Design patterns for exploring and prototyping human-robot interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1439–1448, 2014. (Cited in page 31.)

[249] Thomas Scanlon. *What we owe to each other*. Belknap Press of Harvard University Press, 2000. (Cited in page 16.)

[250] Brian Scassellati. Theory of mind for a humanoid robot. *Autonomous Robots*, 12(1):13–24, 2002. (Cited in page 34.)

[251] Roger C. Schank and Robert P. Abelson. *Scripts, plans, goals, and understanding : an inquiry into human knowledge structures*. Hillsdale, N.J. : Lawrence Erlbaum Associates, 1977. (Cited in page 23.)

[252] Emanuel A Schegloff. The routine as achievement. *Human studies*, 9(2-3):111–151, 1986. (Cited in page 7.)

[253] Emanuel A Schegloff. Word repeats as unit ends. *Discourse Studies*, 13(3):367–380, 2011. (Cited in page 7.)

[254] Emanuel A Schegloff, Gail Jefferson, and Harvey Sacks. The preference for self-correction in the organization of repair in conversation. *Language*, 53(2):361–382, 1977. (Cited in page 25.)

[255] Emanuel A Schegloff and Harvey Sacks. Opening up closings. *Semiotica*, 8(4):289–327, 1973. (Cited in pages 8 and 65.)

[256] Matthias Scheutz, Paul Schermerhorn, and James Kramer. The utility of affect expression in natural language interactions in joint human-robot tasks. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 226–233, 2006. (Cited in pages 33 and 40.)

[257] Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. An overview of the distributed integrated cognition affect and reflection diarc architecture. *Cognitive architectures*, pages 165–193, 2019. (Cited in page 40.)

[258] Stephen Schiffer. *Meaning*. Oxford, Clarendon Press, 1972. (Cited in page 21.)

[259] John Searle. Collective intentions and actions. In Philip R. Cohen Jerry Morgan and Martha Pollack, editors, *Intentions in Communication*, pages 401–415. MIT Press, 1990. (Cited in page 18.)

[260] John R. Searle. *Intentionality: An Essay in the Philosophy of Mind*. Cambridge University Press, 1983. (Cited in page 15.)

[261] Natalie Sebanz, Harold Bekkering, and Günther Knoblich. Joint action: bodies and minds moving together. *Trends in cognitive sciences*, 10(2):70–76, 2006. (Cited in pages 11, 18, 19, and 22.)

[262] Natalie Sebanz and Guenther Knoblich. Prediction in joint action: What, when, and where. *Topics in Cognitive Science*, 1(2):353–367, 2009. (Cited in pages 17 and 22.)

[263] Natalie Sebanz, Günther Knoblich, and Wolfgang Prinz. How two share a task: corepresenting stimulus-response mappings. *Journal of Experimental Psychology: Human Perception and Performance*, 31(6):1234, 2005. (Cited in page 18.)

[264] Julie Shah, James Wiken, Brian Williams, and Cynthia Breazeal. Improved human-robot team performance using chaski, a human-inspired plan execution system. HRI '11, New York, NY, USA, 2011. Association for Computing Machinery. (Cited in page 47.)

[265] Christine Sharbrough. Apa dictionary of psychology, 2015. (Cited in page 11.)

[266] Candace L Sidner and Christopher Lee. Engagement rules for human-robot collaborative interactions. In *IEEE International Conference on Systems, Man and Cybernetics.*, volume 4, pages 3957–3962. IEEE, 2003. (Cited in page 66.)

[267] Roland Siegwart, Kai O Arras, Samir Bouabdallah, Daniel Burnier, Gilles Froidevaux, Xavier Greppin, Björn Jensen, Antoine Lorotte, Laetitia Mayor, Mathieu Meisser, et al. Robox at expo. 02: A large-scale installation of personal robots. *Robotics and Autonomous Systems*, 42(3-4):203–222, 2003. (Cited in page 132.)

[268] Gustavo R. Silva, Leandro B. Becker, and Jomi F. Hübner. Embedded architecture composed of cognitive agents and ros for programming intelligent robots. *IFAC-PapersOnLine*, 53(2):10000–10005, 2020. 21th IFAC World Congress. (Cited in pages 58 and 59.)

[269] Rui Silva, Luís Louro, Tiago Malheiro, Wolfram Erlhagen, and Estela Bicho. Combining intention and emotional state inference in a dynamic neural field architecture for human-robot joint action. *Adaptive Behavior*, 24(5):350–372, 2016. (Cited in page 35.)

[270] Phani-Teja Singamaneni and Rachid Alami. Hateb-2: Reactive planning and decision making in human-robot co-navigation. In *EEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 179–186. IEEE, 2020. (Cited in pages 44 and 148.)

[271] Barbora Siposova and Malinda Carpenter. A new look at joint attention and common knowledge. *Cognition*, 189:260–274, 2019. (Cited in page 20.)

[272] Barbora Siposova, Michael Tomasello, and Malinda Carpenter. Communicative eye contact signals a commitment to cooperate for young children. *Cognition*, 179:192–201, 2018. (Cited in pages 16 and 17.)

[273] Emrah Akin Sisbot and Rachid Alami. A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28(5):1045–1057, 2012. (Cited in page 33.)

[274] Beate Sodian and Susanne Kristen-Antonow. Declarative joint attention as a foundation of theory of mind. *Developmental psychology*, 51(9):1190–1200, 2015. (Cited in page 9.)

[275] D. Sperber and D. Wilson. *Relevance: Communication and Cognition*. Blackwell, 1995. (Cited in page 23.)

[276] Thorsten P Spexard, Marc Hanheide, Shuyin Li, Britta Wrede, et al. Oops, something is wrong-error detection and recovery for advanced human-robot-interaction. 2008. (Cited in page 36.)

[277] Maria Staudte and Matthew W Crocker. Visual attention in spoken human-robot interaction. In *2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 77–84. IEEE, 2009. (Cited in page 33.)

[278] Aaron Steinfeld, Terrence Fong, David Kaber, Michael Lewis, Jean Scholtz, Alan Schultz, and Michael Goodrich. Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*, pages pp. 33–40, Salt Lake City, United States, 2006. (Cited in pages 117 and 169.)

[279] Daniel Szafir, Bilge Mutlu, and Terrence Fong. Communicating directionality in flying robots. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 19–26. IEEE, 2015. (Cited in page 33.)

[280] Aaquib Tabrez, Matthew B. Luebbers, and Bradley Hayes. A survey of mental modeling techniques in human–robot teaming. *Current Robotics Reports*, Aug 2020. (Cited in page 67.)

[281] Leila Takayama, Doug Dooley, and Wendy Ju. Expressing thought: improving robot readability with animation principles. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 69–76, 2011. (Cited in page 33.)

[282] A. Tanevska, G. Rea, F.and Sandini, and A. Sciutti. Towards an Affective Cognitive Architecture for Human-Robot Interaction for the iCub Robot. In *1st Workshop on "Behavior, Emotion and Representation: Building Blocks of Interaction"*, Bielefeld, Germany, October 2017. (Cited in pages 117 and 169.)

[283] Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014. (Cited in page 173.)

[284] Kyle A Thomas, Peter DeScioli, Omar Sultan Haque, and Steven Pinker. The psychology of coordination and common knowledge. *Journal of personality and social psychology*, 107(4):657, 2014. (Cited in page 21.)

[285] Andrea Thomaz, Guy Hoffman, and Maya Çakmak. Computational human-robot interaction. *Foundations and Trends in Robotics*, vol. 4(2-3):pp. 105–223, 2016. (Cited in page 115.)

[286] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: a second-generation museum tour-guide robot. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 3, pages 1999–2005, 1999. (Cited in page 132.)

[287] Deborah Tollefsen. Let's pretend!: Children and joint action. *Philosophy of the Social Sciences*, 35(1):75–97, 2005. (Cited in pages 9, 11, and 15.)

[288] Michael Tomasello. *The cultural origins of human cognition*. Harvard university press, 1999. (Cited in page 20.)

[289] Michael Tomasello and Malinda Carpenter. Shared intentionality. *Developmental science*, 10(1):121–125, 2007. (Cited in page 21.)

[290] Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and brain sciences*, 28(5):675–691, 2005. (Cited in pages 11, 12, 15, 16, 18, 19, and 20.)

[291] Michael Tomasello et al. Joint attention as social cognition. In Chris Moore, Philip J Dunham, and Phil Dunham, editors, *Joint attention: Its origins and role in development*, pages 103–130. Psychology Press, 1995. (Cited in page 20.)

[292] J Gregory Trafton, Laura M Hiatt, Anthony M Harrison, Franklin P Tamborello, Sangeet S Khemlani, and Alan C Schultz. Act-r/e: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, 2(1):30–55, 2013. (Cited in page 39.)

[293] Rudolph Triebel, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers, Michelangelo Fiore, et al. Spencer: A socially aware service robot for passenger guidance and help in busy airports. In *Field and service robotics*, pages 607–622. Springer, 2016. (Cited in page 132.)

[294] Jochen Triesch, Christof Teuscher, Gedeon O Deák, and Eric Carlson. Gaze following: why (not) learn it? *Developmental science*, 9(2):125–147, 2006. (Cited in page 21.)

[295] Raimo Tuomela. *The Importance of Us: A Philosophical Study of Basic Social Notions*. Stanford University Press, 1995. (Cited in page 15.)

[296] Cordula Vesper, Ekaterina Abramova, Judith Bütepage, Francesca Ciardo, Benjamin Crossey, Alfred Effenberg, Dayana Hristova, April Karlinsky, Luke McEllin, Sari R. R. Nijssen, Laura Schmitz, and Basil Wahn. Joint action: Mental representations, shared information and general mechanisms for coordinating with others. *Frontiers in Psychology*, 7:2039, 2017. (Cited in page 18.)

[297] Cordula Vesper, Stephen Butterfill, Günther Knoblich, and Natalie Sebanz. A minimal architecture for joint action. *Neural Networks*, 23(8-9):998–1003, 2010. (Cited in pages 21 and 22.)

[298] Cordula Vesper and Michael J Richardson. Strategic communication and behavioral coupling in asymmetric joint action. *Experimental brain research*, 232(9):2945–2956, 2014. (Cited in page 24.)

[299] J. Waldhart, A. Clodic, and R. Alami. Reasoning on shared visual perspective to improve route directions. In *28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, New Delhi, India, Oct 2019. (Cited in page 146.)

[300] Jules Waldhart, Mamoun Gharbi, and Rachid Alami. A novel software combining task and motion planning for human-robot interaction. In *2016 AAAI Fall Symposium Series*, 2016. (Cited in page 43.)

[301] Mathieu Warnier, Julien Guitton, Séverin Lemaignan, and Rachid Alami. When the robot puts itself in your shoes. managing and exploiting human and robot beliefs. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 948–954. IEEE, 2012. (Cited in page 34.)

[302] Henry M Wellman, David Cross, and Julanne Watson. Meta-analysis of theory-of-mind development: The truth about false belief. *Child development*, 72(3):655–684, 2001. (Cited in page 10.)

[303] Carol Westby and Lee Robinson. A developmental perspective for promoting theory of mind. *Topics in language disorders*, 34(4):362–382, 2014. (Cited in page 9.)

[304] H. Wimmer. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103–128, 1983. (Cited in page 10.)

[305] Daniel M Wolpert, Kenji Doya, and Mitsuo Kawato. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):593–602, 2003. (Cited in page 22.)

[306] Robin Wooffitt and I Hutchby. *Conversation analysis*. Polity, 2008. (Cited in page 25.)

[307] Chen Yu, Matthias Scheutz, and Paul Schermerhorn. Investigating multimodal real-time patterns of joint attention in an hri word learning task. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 309–316. IEEE, 2010. (Cited in page 33.)

[308] Kuanhao Zheng, Dylan F Glas, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Designing and implementing a human–robot team for social interactions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4):843–859, 2013. (Cited in page 27.)

[309] Vittorio A. Ziparo, Luca Iocchi, Pedro U. Lima, Daniele Nardi, and Pier F. Palamara. Petri net plans - A framework for collaboration and coordination in multi-robot systems. *Autonomous Agents and Multi-Agent Systems*, 23(3):344–383, 2011. (Cited in page 50.)

ajouter tous les prenoms pour harmoniser