Overworld: Assessing the geometry of the world for Human-Robot Interaction

Guillaume Sarthou

Overworld: Assessing the geometry of the world for Human-Robot Interaction

Guillaume Sarthou

***Abstract*—For a robot to interact with humans in a given environment, a key need is to understand its environment in terms of the objects composing it, the other agents acting in it, and the relations between all of them. This capability is often called the geometrical situation assessment and is mainly related to spatial reasoning in time.**

In this paper, we present Overworld, a novel lightweight and open-source framework, merging the key features of a decade of research in the domain. It permanently maintains a geometric state of the world from the point of view of the robot by aggregating perceptual information from several sources and reasoning on them to create a coherent world. Furthermore, Overworld implements perspective-taking by emulating the humans' ability to perceive to estimate the state of the world from their perspective. Finally, thanks to a strong link with an ontology framework, it ensures knowledge coherence in the whole robotic architecture. This work is part of a broader effort to develop a complete, stable, and shareable decisional robotic architecture for Human-Robot Interaction.

***Index Terms*—Multi-Modal Perception for HRI; Human-Robot Collaboration; Software Architecture for Robotic and Automation**

I. Introduction

FOR a robot to act on an environment, talk about it, or take decisions in relation to it, one key quality is the ability to reason about it. The robot is not omniscient, indeed, it is limited by the range of its sensors. However, the world is larger than that and can not be limited for example to a single image. This means that the robot has to reason in time, considering the evolution of the percepts. If the robot has no more data on a given object, it can be because it is out of its field of view or because it is now occluded by another one. Considering the presence of humans in the environment, a third explanation could be that the object has been moved by a human. At the difference of robots acting alone, the dynamic of the environment does not only take its origin from the robot's activity. The commonly used paradigm "the perception as an action" used for example in [3] does not hold. The robot has thus to permanently monitor its environment to allow a higher decisional level to react to uncontrollable events.

To facilitate the interaction with humans, making it smoother, more natural, and more efficient, a key capability of the robot at the situation assessment (SA) level is **Visual Perspective Taking** (VPT). It is defined in [18] as *"the ability to predict the visual experience of another agent"*. By doing so, we are able to estimate if another person can see an object or not. This ability is the first step to implement **Theory of Mind** (ToM) being *"the ability to make inferences about what other people believe to be the case in a given situation"* [1]. As shown in [4], [7], or [28], such an ability allows among other things the generation of efficient communication avoiding to consider facts or objects estimated as unknown by the agent the robot interacts with, or at least different from its perspective. See [8] for a survey on VPT use in robotics.

Another important challenge in the context of Human-Robot Interaction (HRI), is to make the robot's knowledge shareable with the humans partners. While the robot perceives the elements of the environment in terms of coordinates, humans are more likely to observe an environment in terms of symbolic relations (e.g. "the plate on the table" or "the knife in your hand"). The robot thus has to be able to extract such relations from its perception of the environment. More than allowing verbal interaction, this ability allows the execution and validation of symbolic task planning where for example the goal defined at the symbolic level could be to put the knife at the right of the plate rather than at a precise coordinate.

In this paper, we present Overworld, a novel lightweight, efficient, and open-source framework, merging the key features of a decade of research in the domain of geometric situation assessment for HRI. The main contributions of this work are an **advanced geometrical reasoning process** independent of the used perception modalities, a strong **link with an ontology** framework coming with **meaning-full symbolic relations extraction**, and **parallel representation of worlds**, all based on the VPT principle.

In §II we briefly discuss related work and how our contribution addresses multiple issues at a time. An overview of the software architecture is then provided in §III before a focus on the reasoning process in §IV and on the symbolic facts computation in §V. Finally, §VI presents results on two different robots performing a benchmark task for HRI and §VII concludes the paper.

II. Related work

In robotics, bridging the gap between sensing and deliberation (as language or task planning) is a common need. However, as explained in [10], it often consists of an ad-hoc integration of processing methods for application-specific or even scenario-specific approaches. In an architecture, the design of a hub for sensor fusion and geometric reasoning aims at proposing a more generic and reusable solution. Such a solution is known as symbol grounding. It establishes and maintains a link between what a system can sense and what

Manuscript received: November 15, 2022; Revised December 14, 2022; Accepted January, 15, 2023.

This paper was recommended for publication by Editor Gentiane Venture upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Agence Nationale de la Recherche DISCUTER project ANR-21-ASIA-0005.

Guillaume Sarthou is with LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France guillaume.sarthou@laas.fr

it can reason about. Coradeschi et al. in [5] present a review of such solutions. Among the reviewed solutions, we can cite the Grounded Situation Model (GSM) [17] which proposes an amodal physical representation of the world used to maintain symbolic beliefs about the objects of the environment and the robot's own body.

To pass from a geometric representation to a symbolic one, spatial reasoning [20] is used. Such abstraction is required for instance for grounded natural language processing [27] during Human-Robot Interaction. For example, [25] presented a spatial reasoner to compute symbolic objects location (e.g `Box isOn Table`) as well as relations between objects and agents (e.g. `Box isVisibleBy Bob`).

To systematize the bridging between sensing and deliberation, Heintz in [9] introduces the notion of knowledge processing middleware and draws some requirements. Among them, we can cite the need for a flexible configuration and reconfiguration or to permit the integration of information from distributed sources. In addition, he highlights that the knowledge process has to be decoupled and asynchronous to a certain degree. Indeed, the direct processing of the sensor data has to be done at a high frequency to allow among others filtering or tracking while the symbolic abstraction could be done at a lower frequency. The processing should thus not be seen as a continuous stream whose tempo is given by the sensors. This is particularly true when several sensors are used at the same time where a full process would be triggered at each new data of each sensor. DyKnow [11] is an implementation of such knowledge processing middleware which has been used in drone applications.

While previous contributions focus only on the robot representation of the world, the integration of such a process for Human-Robot Interaction brings new challenges to be tackled. Using the notion of VPT, Warnier et al. in [29] estimate the beliefs of the humans collaborating with the robot to detect belief divergences and to create an estimate of the humans' knowledge bases. However, the proposed solution is simplistic with the use of a single world being the robot representation and with the use of the assumption that when the human is present in the scene he notices every action on all objects and thus knows their new positions even if he cannot see them. SPARK [19] extends the previous work with more advanced reasoning on objects' positions trying to find hypotheses (i.e. occlusions) to explain why an object is not perceived anymore. However, even if maintaining separate belief bases, SPARK still works on a single world representation not allowing a genuinely independent representation of the human knowledge about the world. With Underworlds [13], Lemaignan et al. proposed a new paradigm to develop a geometrical SA for HRI. Through a principle of cascading, Underworlds allows the representation of multiple parallel representations of the world. Thanks to this, it allows to handle and to maintain truly independent models of the environment for each agent and thus allows the representation of proper false-belief situations. However, Underworlds comes more as a toolkit rather than a software even if example clients are available. As stated by the authors, "it does not provide any intrinsic high-level processing or reasoning capability". In addition, the cascading architecture has the side effect to create a continuous knowledge stream triggered by each new data from each sensor. Nevertheless, it allows fast prototyping as shown in [22] where it has been used to implement simulation-based physics reasoning but not in real-time and without VPT. It is thus mainly used to represent and share multiple parallel representations of the world, to be used by other components.

III. Design and architecture

Overworld aims at gathering the strengths of the previously presented contributions to propose an amodal and efficient solution in addition to provide meaningful geometrical and spatial reasoning capabilities. In this section, we first present used types and the knowledge stream with regard to the robotic architecture Overworld is integrated into. We then give key details on the specifics of its implementation.

A. Used types

Overworld considers two kinds of entities: the objects and the body parts.

Entity: An entity is defined by a unique identifier which can be a "true identifier" if it is a known identifier in the entire robotic architecture or a "hidden identifier" if it has been automatically created in Overworld. For example, when the robot grasps an object, we can perceive that an object exists in the robot's gripper but we can not identify it. In this latter case, we can use a "hidden identifier". An entity also owns a pose, a small history of its poses and a shape. The shape can be defined by basic geometric volume (cube, sphere, or cylinder) or a mesh (visual and collision mesh), in addition to a color, rendering texture and scale.

Object: An object is a specification of an entity. It can be defined as static and can have a mass. In addition, it can own what we call Points of Interest (PoI). Each PoI is a set of points relative to the object which have been used to perceive the object. Using tags to detect an object, the latter will have a PoI per tag and each PoI will be composed of the points of the four corners of the tag. These PoIs will be used to reason about the object's visibility.

Body part: A body part is a specification of an entity. It is defined by a type (e.g. hand, head, torso, or base), a frame name and the agent name it is related to.

Hand: A hand is a specification of a body part in the way that it can hold objects.

Agent: An agent is composed of body parts. Only a head is required to define it. In addition, it has an identifier, a type (robot or human), and a Field of View[1] (FoV).

B. Knowledge stream

Overworld is part of the DACOBOT architecture [24]. It is strongly linked to the software Ontologenius [23], a semantic knowledge base dedicated to HRI. Overworld both updates the knowledge bases and gets from it static knowledge about

[1]For now the FoV only represents a camera placed on the agents' head. Further development should rather describe the agents' sensors and thus not requiring robot's head.

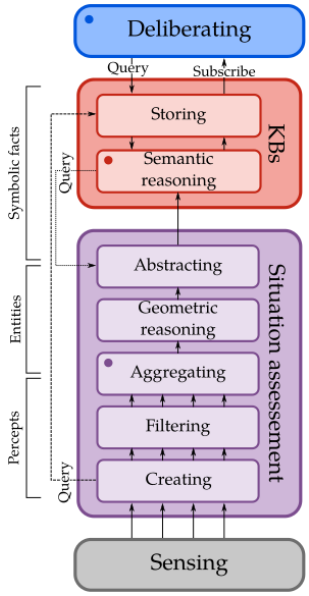

Fig. 1. A partial view of the knowledge stream among the robotic architecture using Overworld. Blocks with a dot of the top-left corner are asynchronous from the block below. The dotted arrow from the Knowledge Bases (KBs) to the Situation Assessment (SA) represents on-demand abstraction. The dashed arrow from the SA to the KBs represents queries about the entities to create.

the entities. This second link allows a uniform representation of knowledge among the entire architecture and eases the configuration by avoiding hard-coded knowledge or the use of multiple configuration files with redundant knowledge.

The knowledge stream represented in Fig. 1 is based on the knowledge processing middleware proposed in [9]. Following the scheme from bottom to top, data coming from sensors or perception processes are sent to Overworld where **percepts** are first created independently for each sensor. Percepts are temporary and possibly incomplete representations of entities of the world. At this level, it can be enriched by static knowledge coming from the knowledge base. Considering a perception process based on tags, from the tags' ids Overworld can query the KB to fetch the internal unique identifiers (e.g. tag 24 corresponds to `table_1`) and the visual and collision 3D models to use, the texture or the color to apply, or the entity's mass if some of this information exists. The position of these percepts can be filtered either to smooth noise or to discard data when the sensor has moved for example.

From there, data have been processed in parallel and in an independent way for each input perception process. An assessment loop (detailed in Sec. IV) periodically pulls all the percepts of all perception processes and aggregates them to create **entities** in the complete representation of the world. For the entities with no data for a few loops, a geometric reasoning process tries to find explanations which can be that the entities are out of the FoV of the sensors or that visual occlusions exist. Once the world is stabilized, spatial reasoning abstracts the representation by generating **symbolic facts** which are sent to the KB.

In an asynchronous way, the KB runs semantic reasoning to deduce new facts from the coming ones (e.g. if `<A, isOn, B>` then `<B, isUnder, A>`) and store them all. Finally, the Supervision or other deliberative processes can either subscribe to patterns of facts (to avoid continuous pulling) or perform direct queries. As detailed in Sec. V, some facts are not required to be computed at each assessment loop and can be computed on demand.

With the described flow, Overworld allows flexible configuration and reconfiguration with the use of the ontology and permits the integration of information from distributed sources as required by [9]. In addition, knowledge processes are decoupled and asynchronous at some levels to avoid over-processing while fitting the frequency requirement of each level of the architecture.

C. Implementation

Overworld is composed of an assessment process per agent it manages. The main one corresponds to the robot while the others correspond to the humans the robot interacts with. Each process is independent of the others and runs in a dedicated thread. Fig. 2 represents Overworld's architecture with two assessment processes.

A set of **perception modules** is used to fetch data from the agent's sensors. These modules are **plugins** of Overworld allowing easy addition of new perceptions capabilities and thus a gain in modularity. A module is dedicated either to perceive objects or body parts. They are responsible for percepts creation and filtering as presented in Sec. III-B. To facilitate its use, Overworld provides three module templates with protection mechanisms. These templates can respectively

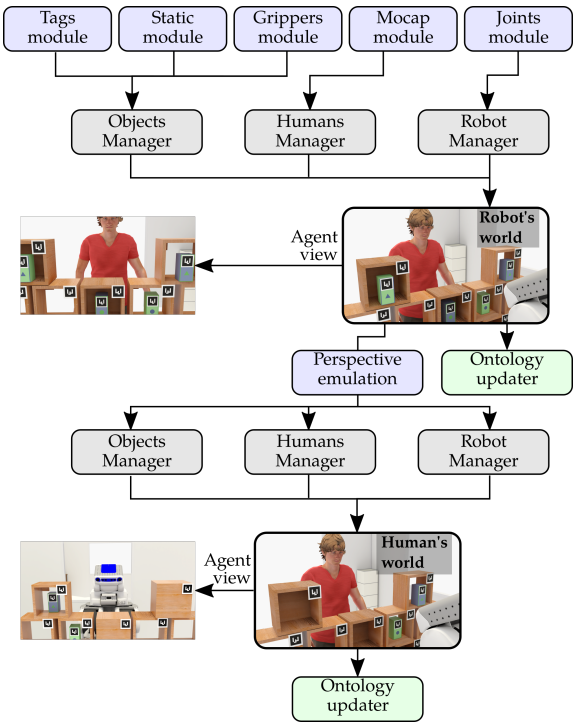

Fig. 2. Schema of Overworld architecture considering one human. Blue boxes represent perception modules as plugins. They feed the grey boxes (being perception managers) with percepts which are then used to create a representation of the world. The human's perceptions modules are emulations of the human ability to perceive in the robot's world.

be used to subscribe to a ROS topic, synchronized ROS topics, and the use of standard C++ callbacks.

While the monolithic structure of SPARK [19] provided good performance, the cascading structure of Underworlds [13] brought powerful modularity. With the plugin solution, Overworld aims at taking the best of both. To select the modules to be used for a given robot or experiment, Overworld considers a **single configuration file** defining the plugins to instantiate as well as their parameters if needed. Considering for example a plugin to perceive objects using the pressure sensors of a robot's gripper, we can describe in the configuration file that this plugin has to be loaded twice, that one is the left gripper with a given pressure threshold and the other for the right gripper with another pressure threshold.

As illustrated in Fig. 2, the perceptions modules are attached to **perception managers**. Overworld has an objects manager, a humans manager, and a robot manager. At each assessment loop (detailed in Sec. IV), the managers fetch the percepts of their modules and run an aggregation algorithm to create entities from the percepts and update them with the new data. This step is required as the same entity could have been perceived by several modules. Considering a robot's gripper perception module, it will provide a percept with an hidden id and a position. At the same time, this object is perceived by a tag module. As both percepts are at the same position with comparable volumes, the manager merges them to create a single entity. When the robot moves its arm and no more perceives the object by its tag, the manager still knows which real entity is manipulated and updates its position thanks to the gripper module.

The managers are also responsible for geometric reasoning. To do so, they rely on the Bullet[2] [6] **physics engine**. Bullet works as a server and allows the creation of multiple parallel worlds. Each assessment process thus owns a world. The three managers of each process update this world with the newly created entities and with the updated data.

Thanks to this 3D representation of the world, the assessment process performs spatial reasoning to compute symbolic facts (detailed in Sec. V) and sends them to the ontology. In addition, for each human perceived by the robot, the robot's assessment process **emulates the humans' ability to perceive**. To do so, it uses the Bullet world to generate a segmentation image of the world from the point of view of each human. These images are then used to feed the perception modules of the humans' assessment processes. At the difference of [19] this emulation and the independent process (and thus reasoning) allows a proper representation and detection of belief divergence between the robot and the humans but also between the humans.

IV. Into the assessment loop

In this section, we focus on the reasoning process performed in the assessment loop. We first detail the geometric reasoning and then present the use of physics simulation.

[2]Overworld uses a custom C++ bullet API based on PyBullet.

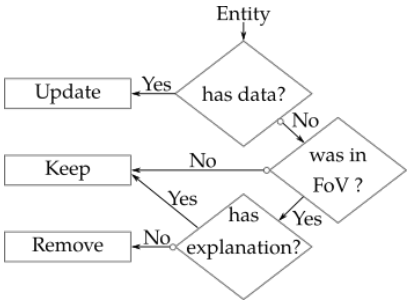

Fig. 3. Diagram of the geometric reasoning algorithm to manage the objects' positions. Explanations can be occlusion or grasp detection.

A. Geometric reasoning

Once the percepts of the modules are fetched, the assessment loop is in charge of updating the agent's representation of the world. To perform this update, once the aggregation is done, we follow for each entity of the world the algorithm represented in Fig. 3.

If the entity has been perceived by one of the modules since the last loop, the entity is created if it was not already and its position is updated. If it has not been perceived, we check if the agent should have been able to perceive it, meaning if the entity is in the agent's FoV. If it is not, we can conclude that this absence of data is normal and we keep the entity's last position. This test is based on the entity's PoIs for those having some (i.e. is a PoI in the FoV?) and is based on the eight corners of its bounding box for those not having PoIs (i.e. is a corner in the FoV?).

In the case the entity is in the agent's FoV, the latter should have perceived it. We thus try to find an explanation for this lack of data. A first explanation could be that the entity is held by an agent. A second explanation could be that the entity is occluded by another one. This latter test differs if the entity has PoIs or not. If it has not, we generate a segmentation image of the scene from the agent's perspective. The entity is thus considered occluded if it does not appear on the image. For the entities having PoIs, we perform a batch of raycasts toward the points composing the PoIs in the FoV. If at least one point of each PoI hits another entity, we conclude that the entity is occluded. Fig. 4 illustrates such a situation with a table perceived with two tags which are occluded by a box. Each tag is a PoI of the table, each having five points. In the image we see two sets of five raycasts, all hitting the box.

When an explanation is found, the entity is kept in the world at its last position. Otherwise, we remove it from the world. The agent thus knows that the entity exists but does not know its current location.

B. Physics simulation

In a dynamic environment and especially with the presence of humans, an object can be moved from a visible position to an occluded one. In the previously studied example, the box had been moved on the tags but what happens if we put the box in front of the block aside and then slide the box backwards? The reasoning process would initially find an explanation but when we would slide the box, in the robot's world the block

would "pass through" the box (as we don't update its pose). After that, no occlusion would be found and the block would be removed. The same would appear when we drop an object in a box. The last perceived would be above the box, and since the cube would have fallen, it would thus not be perceived anymore, and thus be removed from the scene as no occlusion would be detected between the camera and the last position of the cube.

Here the issue is that the algorithm does not rely on the physics of the objects. Physics simulation is commonly used in robotics to predict the effects of an action [2] but not used in real-time to understand what is happening. Nevertheless, as previously explained, [22] has presented a proof of concept for HRI. In our implementation, rather than constantly simulating all the objects, we have chosen to only simulate those for which the search for an explanation is required. This means that we simulate the occluded objects and the ones that we would have been removed otherwise. In the first case, it allows them to react to the movement of other objects like in the example of the slipped box. For the others, it allows to test if their physics could lead them to a position where an occlusion would explain their absence of data. After a few simulation steps, if we still do not find any explanation, then we finally remove them. This addition to the previous reasoning process allows the understanding of more dynamic situations.

V. From geometric to symbolic

We consider two kinds of facts, those computed at each assessment loop and those computed on demand. For all the facts having inverses, Overworld only computes one of them and the other is deduced by the semantic reasoning of the ontology management system. Additional facts could be computed in the future.

A. Continuously computed facts

Overworld continuously computes six facts allowing to trigger higher decisional processes.

`isOnTopOf(Object, Support):` An object is on top of another if the second is defined as a support in the ontology, the lower z (z vertical) coordinate of its Axis Aligned Bounding Box (AABB) is approximately at the same level as the upper z coordinate of the support, and the projection of their AABB on the xy plane are overlapping.

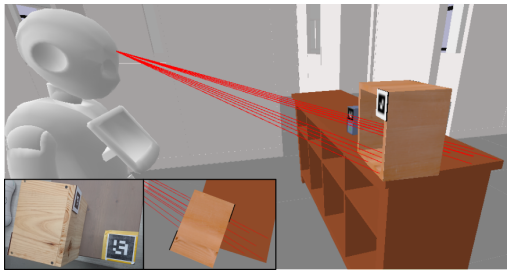

Fig. 4. Third-person view of the robot world in the Bullet GUI. Red lines are a debug visualization of raycasts performed to detect occlusions between the robot's sensor and the PoIs.

`isInContainer(Object, Container):` An object is in another if the second is defined as a container in the ontology, the lower z coordinate of its AABB is approximately contained in the z coordinates of the support, and that the projections of their AABB on the xy plane are overlapping.

`isInHand(Pickable, Hand):` An object is in a hand if it is defined as pickable in the ontology and if it is closer to the hand than a few centimeters. The object is then attached to the hand to follow its movement and is released once it is perceived at a different position from the hand.

`isPerceiving(Agent, Agent):` An agent is perceiving another agent if at least one of the second agent's body parts is in the FoV of the first agent.

`isLookingAt(Agent, Object):` An agent is looking at an object if the object is in the segmentation image generated from the agent's point of view.

`hasHandMovingToward(Hand, Object):` A hand is moving toward an object if the object is in a cone oriented by the mean vector resulting from the N last positions of the hand.

B. On demand facts

Facts with high dynamics and requiring instance computations can be computed on demand. For now, Overworld implements the computation of the facts `isAtLeftOf` and `isInFrontOf`. While [25] implements a simplified version by only considering these relations between the agents and the objects, Overworld comes with a meaningful implementation of these facts, based on the theory proposed in [16].

Following the principle of *"thinking for speaking"* [26], Levelt explains that such relations should be computed in a way that they will be understood. These relations can be computed with regard to three systems. The **absolute** uses the north-south dimension and is neither relative to the speaker's nor the object's coordinate system. The **intrinsic** perspective system uses the objects' intrinsic axis if some exist. For example, in most cultures, we assume a front and upward axis for a chair or a car. We can thus compute spatial relations using their coordinate system. Finally, the **pragmatics** perspective systems use the agents' orientation to compute relations. Following the VPT principle, these relations will thus differ depending on each agent. While the absolute system is rarely used, in most cultures, none of the others is dominant over the others.

Levelt also explains that while converseness (i.e. the inverse of a relation) and transitivity hold in the pragmatics and absolute system, they do not in the intrinsic one. From the experiments presented in [15], we can also notice the principle of canonical orientation explaining that if an object with an intrinsic axis is not in its canonical orientation, it cannot be used as a perspective system. In a simplified definition, an object is in a canonical orientation if its upward axis is aligned with the absolute system's upward axis. In other words, if a chair is placed on one of its sides, it can not be used to refer to another object using the intrinsic system. Finally, as already used in [25] with the relation `isNextTo`, the size of the objects has an impact on the relation computation. Where two

TABLE I
SUMMARY OF COMPARISON BETWEEN OVERWORLD AND UNDERWORLDS OVER TWO BENCHMARKS.

	Situation 1				Situation 2		
	Frequency	Positions before grasping	Positions during grasping	Positions after dropping	Frequency	Positions	Perspective estimation
Underworlds	8 Hz	Fluctuating	Correct	Missing	3.5 Hz	Fluctuating	Correct
Overworld	20 Hz	Correct	Correct	Correct	13 Hz	Correct	Correct

houses spaced of 5 meters are next to each other, two pens also spaced of 5 meters are not next to each other. Because the relations `isAtLeftOf` and `isInFrontOf` are only used for near objects, their sizes have to be considered.

Overworld implements the computation of the relations `isAtLeftOf` and `isInFrontOf` in the intrinsic and pragmatic systems by following all the presented principles. While the intrinsic systems are used identically for all the agents' worlds, the pragmatic system is only related to the world owner. However, as all the worlds feed a dedicated ontology, the robot can still use PT by reasoning on the agents' estimated knowledge base.

To make these facts available to the rest of the architecture in a transparent way, Overworld takes advantage of Ontologenius (the semantic knowledge base) by providing a reasoner in the form of a plugin. The reasoner runs before each query to Ontologenius and if the query involves relations computable by Overworld, it computes them on demand before answering the query[3].

VI. RESULTS

Overworld has been successfully integrated into the DACOBOT robotic architecture and tested on two robots (Pr2 and Pepper) to pass the Director Task [24]. The implementation on Pr2 has been used with object detection based on tags and the robot grippers, and human detection based on external motion capture. The implementation on Pepper has been used only with tags for the objects and computer vision [12] to detect the humans.As illustrated in Fig. 5, the test of the Director Task is interesting to assess the usability of such a SA software as it requires a meaningful and stable computation of facts to generate and understand verbal instructions, as well as precise position management to allow the robot to effectively grasp objects. In addition, to generate efficient communication, this task requires ToM and thus VPT to estimate the others' knowledge about the world.

To go further, we proposed two new benchmark situations[4].

- The first consist of the robot picking a block and then dropping it in a box. The human partner then reverts the box to make the block fall on the table before sliding the box (and thus pushing the block) until this latter falls on the ground. This first situation evaluates the percepts' aggregation and the geometric reasoning with physics simulation. Indeed, initially, the robot perceives the block with a tag modality. Once picked, the tag becomes hidden and the robot has to use its gripper sensors to detect that the object is in the gripper. Finally, for the rest of the manipulation, the block is no more perceived and its position has to be estimated with geometric reasoning with physics simulation.
- The second situation consists of two humans around the DT setup. One is on the robot's side while the other is in front of the robot. The second human then moves to the robot side to discover a previously hidden block. This second situation mainly evaluates the software scalability (with 13 objects and 2 humans) as well as the VPT. This means that three worlds have to be maintained simultaneously and that two perspectives have to be computed.

Both Overworld and Underworlds have been tested against these benchmarks using a rosbag to provide them with identical inputs. As Underworlds is a toolkit not implementing high-level reasoning, we used the implementation developed to prototype Overworld. This implementation is based on an existing one [21] proposed by Underworlds's authors to ensure a correct software architecture. The tested Underworlds implementation integrates reasoning on objects without physics simulation. We have to note that Underworlds messages exchanged between worlds had to be modified to support PoI. The same symbolic facts were computed by both software. SPARK had not been tested as it is no longer available with features shown in its related works. As explained in [14], it was able to run at 10Hz without physics simulation.

The first benchmark is passed by Overworld at 20Hz with a correct object position estimation and VPT. Underworlds runs it at 8Hz with several objects disappearances due to difficulties to synchronize data. Furthermore, once picked by the robot, the block position was no more estimated. The second benchmark is passed by Overworld and Underworlds respectively at 13Hz and 3.5Hz. Both positions' estimations were correct as the robot is only a spectator of a static situation and uses a single perception capability. We see that Overworld is at least twice

[3]For efficiency issues, Overworld is not requested to recompute the facts under a given duration corresponding to its assessment loop.

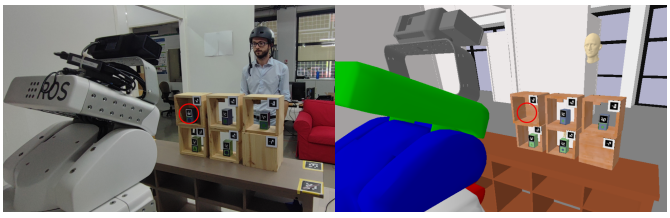

Fig. 5. A third-person view of the real setup from the robot side (left image) and the corresponding estimation of the human's representation of the situation (right image). The robot has estimated that the human cannot know that a block exists in the left-top box (red circle).

[4]Available at https://gitlab.laas.fr/gsarthou/overworld_benchmark

faster as Underworlds and is less impacted by scaling up the complexity. These results are summarized in Tab. I.

Additional results are available in the attached video[5].

VII. Conclusion

In this paper, we have presented Overworld, our geometrical SA system for HRI. We exposed how its architecture and its integration into a robotic architecture allow to build a coherent knowledge processing middleware with a strong and bidirectional link to an ontology easing the system configuration and adding semantic reasoning on top of the spatial reasoning. The configurable plugin system has been shown to facilitate the system extension with new perception capabilities and to allow its use on several robotic platforms. For the community to use it, some can either implement their own perception modules or take existing ones to match their robot perception capabilities and then take advantage of all Overworld reasoning mechanisms.

With the current contribution, we have also highlighted advanced reasoning capabilities both in terms of geometrical reasoning to create a coherent world and in terms of meaningful symbolic relations extractions. Finally, we have demonstrated that Overworld can maintain several parallel representations of the world to implement ToM through the use of VPT. In addition to the ability to emulate the human ability to perceive, we have shown that Overworld allows a proper false-belief representation.

Even if we have successfully gathered decades of research on SA for HRI, Overworld could be brought far ahead in future work. Where some reasoning processes focus on the agents' FoV underlying the use of a single visual sensor, we want to integrate a fine representation of the agents' sensors and integrating them into the reasoning process. Such addition will lead to a finer representation of the used modules to perceive a given entity allowing more precise reasoning.

For now, Overworld uses the convenient and efficient 4x4 transformation matrix to represent the entities' position in the world. However, we could gain with explicit management of uncertainties especially when the same entity is perceived by several perception modules at the time. We thus plan to integrate a position representation with a covariance matrix for the next developments.

In an effort to share this work with the robotics community, a website[6] and the code[7] are available.

Acknowledgement

The author wants to thank Guilhem Buisan for his work on the implementation of this software.

References

[1] S. Baron-Cohen, A. M. Leslie, and U. Frith, "Does the autistic child have a "theory of mind"?" *Cognition*, 1985.

[2] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. K. Bozcuoğlu, and G. Bartels, "Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents," in *IEEE ICRA*, 2018.

[3] M. Beetz, L. Mösenlechner, and M. Tenorth, "Cram—a cognitive robot abstract machine for everyday manipulation in human environments," in *IEEE/RSJ IROS*, 2010.

[4] G. Buisan, G. Sarthou, A. Bit-Monnot, A. Clodic, and R. Alami, "Efficient, situated and ontology based referring expression generation for human-robot collaboration," in *IEEE RO-MAN*, 2020.

[5] S. Coradeschi, A. Loutfi, and B. Wrede, "A short review of symbol grounding in robotic and intelligent systems," *KI-Künstliche Intelligenz*, 2013.

[6] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[7] F. I. Doğan, S. Gillet, E. J. Carter, and I. Leite, "The impact of adding perspective-taking to spatial referencing during human–robot interaction," *Robotics and Autonomous Systems*, 2020.

[8] N. Gurney and D. V. Pynadath, "Robots with theory of mind for humans: A survey," in *IEEE RO-MAN*, 2022.

[9] F. Heintz, "Dyknow: A stream-based knowledge processing middleware framework," Ph.D. dissertation, Linköping University Electronic Press, 2009.

[10] F. Heintz, J. Kvarnström, and P. Doherty, "Knowledge processing middleware," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2008.

[11] ——, "Bridging the sense-reasoning gap: Dyknow–stream-based middleware for knowledge processing," *Advanced Engineering Informatics*, 2010.

[12] V. Khalidov and J.-M. Odobez, "Real-time multiple head tracking using texture and colour cues," Idiap, Tech. Rep. Idiap-RR-02-2017, 2017.

[13] S. Lemaignan, Y. Sallami, C. Wallbridge, A. Clodic, T. Belpaeme, and R. Alami, "Underworlds: Cascading situation assessment for robots," in *IEEE/RSJ IROS*, 2018.

[14] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, and R. Alami, "Artificial cognition for social human–robot interaction: An implementation," *Artificial Intelligence*, 2017.

[15] W. J. Levelt, "Some perceptual limitations on talking about space," in *Limits in perception*. CRC Press, 1984.

[16] ——, "Perspective taking and ellipsis in spatial descriptions," *Language and space*, 1999.

[17] N. Mavridis and D. Roy, "Grounded situation models for robots: Bridging language, perception, and action," in *AAAI-05 workshop on modular construction of human-like intelligence*, 2005.

[18] P. Michelon and J. M. Zacks, "Two kinds of visual perspective taking," *Perception & psychophysics*, 2006.

[19] G. Milliez, M. Warnier, A. Clodic, and R. Alami, "A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management," in *IEEE RO-MAN*, 2014.

[20] J. O'Keefe, "The spatial prepositions in english, vector grammar, and the cognitive map theory," *Language and space*, 1999.

[21] Y. Sallami, "Development psychology inspired models for physical and social reasoning in human-robot joint action," Ph.D. dissertation, Université Toulouse 3 Paul Sabatier, 2021.

[22] Y. Sallami, S. Lemaignan, A. Clodic, and R. Alami, "Simulation-based physics reasoning for consistent scene estimation in an hri context," in *IEEE/RSJ IROS*, 2019.

[23] G. Sarthou, A. Clodic, and R. Alami, "Ontologenius: A long-term semantic memory for robotic agents," in *IEEE RO-MAN*, 2019.

[24] G. Sarthou, A. Mayima, G. Buisan, K. Belhassein, and A. Clodic, "The director task: a psychology-inspired task to assess cognitive and interactive robot architectures," in *IEEE RO-MAN*, 2021.

[25] E. A. Sisbot, R. Ros, and R. Alami, "Situation assessment for human-robot interactive object manipulation," in *IEEE RO-MAN*, 2011.

[26] D. I. Slobin, "Thinking for speaking," in *Annual Meeting of the Berkeley Linguistics Society*, 1987.

[27] S. Tellex, "Natural language and spatial reasoning," Ph.D. dissertation, Massachusetts Institute of Technology, 2010.

[28] J. G. Trafton, N. L. Cassimatis, M. D. Bugajska, D. P. Brock, F. E. Mintz, and A. C. Schultz, "Enabling effective human-robot interaction using perspective-taking in robots," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2005.

[29] M. Warnier, J. Guitton, S. Lemaignan, and R. Alami, "When the robot puts itself in your shoes. managing and exploiting human and robot beliefs," in *IEEE RO-MAN*, 2012.

[5] https://youtu.be/LUkjts8UacI

[6] https://sarthou.github.io/overworld

[7] https://github.com/sarthou/overworld