Second Language Tutoring using Social Robots

Project No. 688014

L2TOR
Second Language Tutoring using Social Robots

Grant Agreement Type: Collaborative Project
Grant Agreement Number: 688014

D4.3 Input module for storytelling domain

Due Date: **30/09/2018**
Submission Date: **30/10/2018**

Start date of project: **01/01/2016** Duration: **36 months**

Organisation name of lead contractor for this deliverable: **Plymouth University**

Responsible Person: **Tony Belpaeme** Revision: **1.0**

Project co-funded by the European Commission within the H2020 Framework Programme		
Dissemination Level		
PU	Public	**PU**
PP	Restricted to other programme participants (including the Commission Service)	
RE	Restricted to a group specified by the consortium (including the Commission Service)	
CO	Confidential, only for members of the consortium (including the Commission Service)	

Contents

Executive Summary

This document contains an update on the changes to the input modalities since D4.2. As the two domain (the space domain and the story-telling domain) have been merged into a single learning experience, the changes to the input modalities have been relatively modest compared to the software we used in 2017. Considerable effort has however been invested in designing a pleasant learning experience for the young learners, and on evaluating how people (both children and adults) perceive and communicate about spatial relations, with a specific focus on how spatial relations can be expressed and communicated between people and social robots.

Principal Contributors

The main authors of this deliverable are as follows (in alphabetical order):

Tony Belpaeme, Ghent University/Plymouth University
Christopher Wallbridge, Plymouth University

Revision History

Version 1.0 (T.B. 10-10-2018) First draft of report.

Version 1.1 (C.W. 30-10-2018)

Version 1.2 (T.B. 15-11-2018) Final version of report.

1 Deliverable Outline

This deliverable is the final deliverable of WP4 (Multimodal input processing) and reports on the efforts in building autonomous perception for social robotics, specifically for social robots used in tutoring applications.

2 Background

The robot has a number of components which aid in social signal processing. Some components were thoroughly evaluated but not used in the final robot tutoring system, which was evaluated in the large-scale L2TOR study. Automated speech recognition, for example, was deemed immature for children's speech (Kennedy et al., 2017) and was not further explored as an input modality for the system. Other input modalities, though functional, were not included in the final system for the large-scale evaluation, as their contribution to the autonomous functioning of the system was not needed. The Voice Activity Detection, face recognition, and emotion recognition components were not used in the large-scale evaluation system. Components such as face detection, motion detection, tablet input and to a limited extent speech recognition (for simple utterances, such as yes/no replies) were used in the final system.

This deliverable reports on work in T4.3: Head pose, gaze tracking and gesture (M13-M36), T4.4: Emotion and affect recognition (M13-M24), T4.5: Tablet input (period M13-M36) and T4.6: Environment processing (period M1-M36).

3 Completed work

3.1 Head pose, gaze tracking and gesture (T4.3)

Gaze tracking is still relatively difficult to achieve using low-res camera images, especially in child-robot interaction. We did however explore the potential of using head tracking as a proxy for gaze tracking using the OpenPose software. OpenPose proved to be particularly effective at tracking skeletons, including head pose, in low quality data from dynamic child interaction (Lemaignan et al., 2018a). However, it should be noted that head tracking is a poor proxy for gaze tracking, which was shown by partner PLYM in a related project (Kennedy et al., 2015).

Gesture tracking was used in two evaluation studies in L2TOR (de Wit et al., 2017; Vogt et al., 2017), and relied on skeleton tracking using the Kinect SDK and the Microsoft Kinect One sensor.

3.2 Emotion and affect recognition (T4.4)

Reading emotions or affect has typically been approached by using computer vision to read faces. This has traditionally relied on machine learning, which has been trained on datasets containing adults faces, often expressing acted out emotions. Because of that, the emotion reading software which is available commercially, or which is available as open source output from research projects, does not work well for highly dynamic contexts in which young users are observed using a low-quality camera. In addition, existing methods only report a few categories of affect (often only Ekman's six basic emotions) in a winner-takes-all fashion. This lack of subtlety makes it less appropriate for adaptively changing the interaction to respond to changes in the user's emotion.

We explored how motion data, as captured by a RGB or RGBD camera, could be used to read emotion, as well as other internal states (Bartlett et al., 2018). To this end we collected a large dataset

consisting of 45 hours of annotated interactions between children and between a child and robot. The PInSoRo dataset, containing tracked skeletons of the videos and annotations, has been made publicly available for data-driven approaches to social Human-Robot Interaction (Lemaignan et al., 2018a, see appendix).

At Bielefeld University, we tried to build a training set for an affect classifier based on Kinect recordings, but unfortunately the inter-rater agreement was too low between the coders (who were trained as teachers) to allow for the training of a classifier. The affective expressions of children during learning are very subtle and context dependant, making it hard, even for trained teachers, to agree on what the affective state is of the child. For the experiments in which the tutoring system adapts the difficulty (see WP5), we relied on a single Wizard to recognise emotions during the interaction, and the system adapted its responses based on this input. While the reading of the affective state might not be correct, it is because of the use of a single use still consistent.

3.3 Tablet input (T4.5)

The main input modality to the robot is the touch-screen tablet, which sits between child and the robot (see figure 1). The tablet is used to display educational content, implemented as animated scenes, and to record responses by the child. This on the one hand gives feedback on the child's learning and performance, but also allows a view on the child's responsiveness, which indirectly informs the system about the child's engagement. Timing information can be used to adapt the learning experience. Ramachandran et al. (2017) for example introduced small breaks when the evolution of the response time decreased beyond a pre-set threshold, and showed it to be an effective strategy to re-engage children with a subsequent positive impact on learning.

Figure 1: The evaluation setup, with a Microsoft Surface tablet being used to display educational content and to collect responses from the young learner.

3.4 Environment processing (T4.6)

Environment processing is a catch-all term for processing social cues and environment cues not directly related to the dyadic interaction between the child and the robot. While being aware of the wider social environment could be useful for some HRI applications, in L2TOR we have not invested in this due to

the constrained nature of the dyadic interaction which does not require the system to be aware of social others.

We did however continue to develop the UNDERWORLDS system which is responsible for reading the spatial environment of objects. It is a lightweight software system for spatio-temporal situation assessment. It builds an internal representation of physical objects, which could exist in the real world or in a simulation environment, and calculates visibility for actors in the environment and spatial relations between objects (Lemaignan et al., 2018b).

In L2TOR, UNDERWORLDS is used to support the spatial reasoning of the robot when teaching spatial language to the child. It reads the 3D simulated environment, and can build representations such as "GIRL NEXT-TO SWING" which in turn generates the output "the girl is next to the swing". It respond to changes, so if the child moves objects or actors on screen, UNDERWORLD updates its internal representation and generates new utterances.

References

Bartlett, M., Belpaeme, T., Thill, S., Edmunds, C. E. R., and Lemaignan, S. (2018). What can you see? identifying cues on internal states from the kinematics of natural social interactions. In *IDC - Workshop 'The Near Future of Childrens Robotics' 2018*.

de Wit, J., Schodde, T., Willemsen, B., Bergmann, K., de Haas, M., Kopp, S., Krahmer, E., and Vogt, P. (2017). Exploring the effect of gestures and adaptive tutoring on childrens comprehension of l2 vocabularies. In *Proceedings of the Workshop R4L at ACM/IEEE HRI 2017*.

Kennedy, J., Baxter, P., and Belpaeme, T. (2015). Head pose estimation is an inadequate replacement for eye gaze in child-robot interaction. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, pages 35–36. ACM.

Kennedy, J., Lemaignan, S., Montassier, C., Lavalade, P., Irfan, B., Papadopoulos, F., Senft, E., and Belpaeme, T. (2017). Child speech recognition in human-robot interaction: evaluations and recommendations. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 82–90. ACM.

Lemaignan, S., Edmunds, C. E., Senft, E., and Belpaeme, T. (2018a). The pinsoro dataset: Supporting the data-driven study of child-child and child-robot social dynamics. *PloS one*, 13(10):e0205999.

Lemaignan, S., Sallami, Y., Wallbridge, C., Clodic, A., Belpaeme, T., and Alami, R. (2018b). Underworlds: Cascading situation assessment for robots.

Ramachandran, A., Huang, C.-M., and Scassellati, B. (2017). Give me a break!: Personalized timing strategies to promote learning in robot-child tutoring. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 146–155. ACM.

Vogt, P., de Wit, J., Schodde, T., Willemsen, B., Bergmann, K., de Haas, M., Kopp, S., and Krahmer, E. (2017). Adaptation and gestures in second language tutoring using social robots.

A Annex Descriptions

A.1 Bartlett et al. (2018) What Can You See? Identifying Cues on Internal States from the Kinematics of Natural Social Interactions

Bibliography - Bartlett, M., Belpaeme, Thill,S., Edmunds, C. E. R., Lemaignan, S. (2018) What Can You See? Identifying Cues on Internal States from the Kinematics of Natural Social Interactions. In *IDC - Workshop 'The Near Future of Childrens Robotics' 2018*.

Abstract - One goal of research on child-robot interactions is to enable robots to autonomously adapt to a childs behaviour in applications such as tutoring and therapeutic settings, for example, adapting to a childs learning or therapeutic needs. This requires robots to track and interpret the internal states of human interaction partners. Studying how humans are able to infer the internal states of others can guide research aiming to endow robots with this skill. Researchers in the fields of psychology and Human-Robot Interaction (HRI) have identified that humans use information such as observed motor activity and contextual information to judge the internal states (e.g. intentions) of others. To design robots able to track the internal states of children it is necessary to first determine what internal-state cues are available from the different sources of information within a social scene, and thereby determine what data are sufficient for internal-state-reading in these scenarios. It is also important to consider the quality and availability of data.

Relation to WP - This work directly contributes to Tasks T4.1-T4.3.

A.2 Lemaignan et al. (2018) The PInSoRo dataset: Supporting the data-driven study of child-child and child-robot social dynamics

Bibliography - Lemaignan, S., Edmunds, C.E.R., Senft, E., Belpaeme, T. (2018) *PLOS One*, 13(10):e0205999. DOI: 10.1371/journal.pone.0205999 .

Abstract - The study of the fine-grained social dynamics between children is a methodological challenge, yet a good understanding of how social interaction between children unfolds is important not only to Developmental and Social Psychology, but recently has become relevant to the neighbouring field of Human-Robot Interaction (HRI). Indeed, child-robot interactions are increasingly being explored in domains which require longer-term interactions, such as healthcare and education. For a robot to behave in an appropriate manner over longer time scales, its behaviours have to be contingent and meaningful to the unfolding relationship. Recognising, interpreting and generating sustained and engaging social behaviours is as such an importantand essentially, openresearch question. We believe that the recent progress of machine learning opens new opportunities in terms of both analysis and synthesis of complex social dynamics. To support these approaches, we introduce in this article a novel, open dataset of child social interactions, designed with data-driven research methodologies in mind. Our data acquisition methodology relies on an engaging, methodologically sound, but purposefully underspecified free-play interaction. By doing so, we capture a rich set of behavioural patterns occurring in natural social interactions between children. The resulting dataset, called the PInSoRo dataset, comprises 45+ hours of hand-coded recordings of social interactions between 45 child-child pairs and 30 child-robot pairs. In addition to annotations of social constructs, the dataset includes fully calibrated video recordings, 3D recordings of the faces, skeletal informations, full audio recordings, as well as game interactions.

A.3 Lemaignan et al. (2018) Underworlds: Cascading Situation Assessment for Robots

Bibliography - Lemaignan, S., Sallami, Y., Wallbridge, C.D., Clodic, A., Belpaeme, T., Alami, R. to be published in *Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*

Abstract - We introduce Underworlds, a novel lightweight framework for *cascading spatio-temporal situation assessment* in robotics. Underworlds allows programmers to represent the robot's environment as real-time distributed data structures, containing both scene graphs (for representation of 3D geometries) and timelines (for representation of temporal events). Underworlds supports *cascading* representations: the environment is viewed as a set of *worlds* that can each have different spatial and temporal granularities, and may inherit from each other. Underworlds also provides a set of high-level client libraries and tools to introspect and manipulate the environment models.

This article presents the design and architecture of this open-source tool, and explores some applications, along with examples of use.

Relation to WP- This work contributes to Task T4.6.

A Annexes

What Can You See? Identifying Cues on Internal States from the Kinematics of Natural Social Interactions

Madeleine Bartlett
CRNS, University of Plymouth
Plymouth, PL4 8AA, UK
madeleine.bartlett@
plymouth.ac.uk

C E R Edmunds
CRNS, University of Plymouth
Plymouth, PL4 8AA, UK
charlotte.edmunds@
plymouth.ac.uk

Séverin Lemaignan
CRNS, University of Plymouth
Plymouth, PL4 8AA, UK
severin.lemaignan@
plymouth.ac.uk
and
BRL, University of the West of England
Bristol, BS16 1QY
severin.lemaignan@brl.ac.uk

Tony Belpaeme
CRNS, University of Plymouth
Plymouth, PL4 8AA, UK
tony.belpaeme@
plymouth.ac.uk
and
ID Lab – imec
University of Ghent, Belgium
tony.belpaeme@ugent.be

Serge Thill
CRNS, University of Plymouth
Plymouth, PL4 8AA, UK
and
Interaction Lab
University of Skövde
541 28 Skövde, Sweden
serge.thill@
plymouth.ac.uk

Introduction

One goal of research on child-robot interactions is to enable robots to autonomously adapt to a child's behaviour in applications such as tutoring [4] and therapeutic settings [5], for example, adapting to a child's learning or therapeutic needs. This requires robots to track and interpret the internal states of human interaction partners. Studying how humans are able to infer the internal states of others can guide research aiming to endow robots with this skill. Researchers in the fields of psychology and Human-Robot Interaction (HRI) have identified that humans use information such as observed motor activity [7] and contextual information [3] to judge the internal states (e.g. intentions) of others. To design robots able to track the internal states of children it is necessary to first determine what internal-state cues are available from the different sources of information within a social scene, and thereby determine what data are sufficient for internal-state-reading in these scenarios. It is also important to consider the quality and availability of data.

Here we discuss the use of skeletal data which is often easily obtained and, when provided by tools such as OpenPose [9] which deals well with occlusions, of high quality. Specifically, we propose a methodology for identifying what humans gain from the kinematics of a child-child social interaction. The findings from studies based on this method-

ology could act as a baseline for what an artificial system can be expected to glean from such data.

Background

Studies examining the mirror neuron system (MNS) found in primates and humans indicate that humans use observed kinematics to make inferences about the observed actor [7, 3]. Broadly speaking, one can identify two types of theory which describe this process. The first type of theory proposes that recognition is a result of an observer mapping the observed kinematics onto their own motor system which allows them to simulate a representation of the intentions driving the observed action [7]. Importantly, this mechanism uses only kinematic information to infer intention. One problem with this account is that humans are able to deal with situations where the same action could be driven by different intentions (e.g. grasping a cup to drink, or to clean it) [3]. A second school of thought incorporates processing of contextual cues (e.g how dirty the cup is) into the MNS whereby identical actions driven by different intentions can be differentiated [2]. Evidence supporting the argument that contextual information influences intention-reading comes from Iacoboni et al. [3] who asked participants undergoing an fMRI scan to watch video clips of a reach-to-grasp action. The information available in the videos was manipulated with three conditions: (1) action embedded in context, (2) action without context, (3) context alone. These were nested within two further conditions such that the same action was driven by one of two intentions. Iacoboni et al. found that participants' neural activity was reliably different between the two intention conditions, and that the MNS was most active when the action was embedded in context. This suggests that intention recognition involves integrating both contextual and kinematic information.

The successful design and training of artificial internal-state-reading systems for child-robot interactions requires that a mapping between the inputs (e.g. a child's posture) and outputs (e.g. a child's internal state) is available. For this, it is important that we identify what internal-state information is available in the different data sources. This can be achieved by assessing what inferences humans are able to make from, for example, the kinematics and dynamics of a social scene (like on Fig. 1, right). One way to do this is by using point-light displays where the position and movements of an actors joints are denoted on an otherwise blank display. Studies using this method have already shown that humans are able to recognise features such as gender [1] and intention [8] from these types of stimuli. HRI researchers can use these findings to define what outputs an artificial system should be able to produce given kinematic data.

However, one key limitation of these studies is that the stimuli used are often artificially produced, e.g. by creating simulated motions in the point-light displays (e.g. [1]), or by filming actors performing the actions in an artificial setting, (e.g. [3, 8]). Whilst this allows researchers to demonstrate that internal-state information is available in kinematics, it does not provide us with insight into what humans can infer from the kinematics of real-world social interactions. Additionally, for child behaviour specifically, creating an artificial dataset may be more challenging, for example, due to variations in cognitive ability with age. Obtaining data from natural interactions is therefore potentially easier and more ecologically valid. The rest of this paper discusses a methodology aimed at identifying what internal-state information humans can glean from only the kinematic information available in a naturalistic child-child social interaction.

Figure 1: Original video clip vs. skeletal only data

Proposed Methodology

Predictions and Design: The proposed study aims to examine what information is available in the kinematics of a naturalistic child-child social interaction. To do this participants will either be shown the original or skeletal videos of real interactions (Fig. 1) and then asked questions about the videos. There will be two questioning conditions where participants are either asked only open-ended questions, or are also asked specific questions. Participants' responses following the original clips will be compared to those following the skeletal videos. Whilst we expect that participants will produce less detailed descriptions following skeletal compared to the original videos, we do expect participants to detect important features from the skeletal videos which would be useful to a robot system, such as the affective valence of the interaction, actions being performed, and the nature of the relationship between the agents.

The proposed study will have a 2 (open-ended/specific questions)$\times$2 (original/skeletal videos) design. Both conditions will be implemented between-subjects. Video presentation order will be fully random to control for ordering effects. Participants will be recruited from a crowd-sourcing platform.

Stimuli and Materials: To obtain naturalistic stimuli the proposed study will utilise videos of child-child pairs playing a game on a touch-screen table top from the PInSoRo dataset [6], made openly available by our group[1]. Short clips of child-child interactions approximately 30 seconds long will be extracted from the videos, each containing different social and interaction events (e.g. turn-taking, a disagreement). To isolate the kinematic information from contextual cues for the skeletal video condition, the OpenPose library [9] is used. It jointly detects human body, hand, and facial landmarks.

After each clip participants will be asked questions about the interaction. There will be two questioning conditions such that half of the participants are asked a single open-ended question following each video: "*Describe what you have just seen in the video*". This style of questioning reduces the risk of "leading questions", allowing us to explore what participants gain from the video without guidance. However, it is often difficult to analyse open responses and respondents may not provide enough detail to reflect their achieved level of insight on features-of-interest. To deal with these limitations half of the participants will be given the same open-ended question, then a series of specific questions which will guide respondents to discuss details of interest to the researcher in a quantifiable manner. The specific questions consist of multiple-choice and Likert scale questions such as "*What is the relationship between these characters: Friends/Neutral/Unfriendly?*" and "*Please rate how cooperative each character was: 1 = not cooperative at all, 10 = very cooperative*". Participants in the specific questioning condition will also be given a final open-ended question on each trial asking "*Did you notice anything else in the video?*".

[1] https://freeplay-sandbox.github.io

Conclusion

The proposed method aims to provide insight into what internal-state information humans are able to glean from kinematic data, with a focus on social situations. The findings of such a study have the potential to guide the design of artificial internal-state-reading systems by providing an expectation of what inferences/outputs the system should be able to draw from the data. Specifically, we plan to apply this knowledge to inform the design of an automatic classifier of social interactions. Whilst the study discussed focuses on kinematic data for internal-state reading in naturalistic interactions with children, this methodology could easily be adapted to examine the information available in a variety of data sources independently of other inputs. We argue that conducting this type of study is an important step when developing robot systems as it can help to streamline the process and provide more direct empirical support for the use of particular data types as inputs to the robot system. For example, by examining how humans recognise when a child is having difficulty with a task or activity, robot tutors could be made able to identify when assistance needs to be provided to a student during a lesson.

Acknowledgements

This work has been funded by the EU H2020 Marie Skłodowska-Curie Actions project DoRoThy (grant 657227) and the EU FP7 project DREAM project (www.dream2020.eu, grant no. 611391)

REFERENCES

1. Mather G and Murdoch L. 1994. Gender discrimination in biological motion displays based on dynamic cues. *Proc. R. Soc. Lond. B* 258, 1353 (1994), 273–279.
2. Kilner JM, Friston KJ, and Frith CD. 2007. Predictive coding: an account of the mirror neuron system. *Cognitive processing* 8, 3 (2007), 159–166.
3. Iacoboni M, Molnar-Szakacs I, Gallese V, Buccino G, and Mazziotta J C. 2005. Grasping the intentions of others with one's own mirror neuron system. *PLoS Biology* 3, 3 (2005), 0529–0535.
4. Baxter P, Ashurst E, Read R, Kennedy J, and Belpaeme T. 2017. Robot education peers in a situated primary school study: Personalisation promotes child learning. *PloS one* 12, 5 (2017), e0178126.
5. Esteban PG, Baxter P, Belpaeme T, Billing E, Cai H, Cao HL, Coeckelbergh M, Costescu C, David D, De Beir A, and Fang Y. 2017. How to build a supervised autonomous system for robot-enhanced therapy for children with autism spectrum disorder. *Paladyn, Journal of Behavioral Robotics.* 8, 1 (2017), 18–38.
6. Lemaignan S, Edmunds C, Senft E, and Belpaeme T. 2017. The Free-play Sandbox: a Methodology for the Evaluation of Social Robotics and a Dataset of Social Interactions. *arXiv preprint arXiv:1712.02421.* (2017).
7. Gallese V, Fadiga L, Fogassi L, and Rizzolatti G. 1996. Action recognition in the premotor cortex. *Brain* 119, 2 (1996), 593–609.
8. Manera V, Becchio C, Cavallo A, Sartori L, and Castiello U. 2011. Cooperation or competition? Discriminating between social intentions by observing prehensile movements. *Experimental Brain Research* 211, 3-4 (2011), 547–556.
9. Cao Z, Simon T, Wei SE, and Sheikh Y. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *CVPR*.

PLOS ONE

RESEARCH ARTICLE

The PInSoRo dataset: Supporting the data-driven study of child-child and child-robot social dynamics

Séverin Lemaignan[1]*, **Charlotte E. R. Edmunds**[2], **Emmanuel Senft**[2], **Tony Belpaeme**[2,3]

1 Bristol Robotics Lab, University of the West of England, Bristol, United Kingdom, **2** Centre for Robotics and Neural Systems, University of Plymouth, Plymouth, United Kingdom, **3** IDLab – imec, Ghent University, Ghent, Belgium

* severin.lemaignan@brl.ac.uk

OPEN ACCESS

Citation: Lemaignan S, Edmunds CER, Senft E, Belpaeme T (2018) The PInSoRo dataset: Supporting the data-driven study of child-child and child-robot social dynamics. PLoS ONE 13(10): e0205999. https://doi.org/10.1371/journal.pone.0205999

Editor: Michael L. Goodman, University of Texas Medical Branch at Galveston, UNITED STATES

Received: June 5, 2018

Accepted: October 4, 2018

Published: October 19, 2018

Data Availability Statement: The dataset is freely available to any interested researcher. Due to ethical and data protection regulations, the dataset is however made available in two forms: - a public, Creative Commons licensed, version that does not include any video material of the children (no video nor audio streams), and hosted on the Zenodo open-data platform: https://zenodo.org/record/1043508. - the complete version that includes all video streams is freely available as well, but interested researchers must first fill a data protection form. The detail of the procedure are

Abstract

The study of the fine-grained social dynamics between children is a methodological challenge, yet a good understanding of how social interaction between children unfolds is important not only to Developmental and Social Psychology, but recently has become relevant to the neighbouring field of Human-Robot Interaction (HRI). Indeed, child-robot interactions are increasingly being explored in domains which require longer-term interactions, such as healthcare and education. For a robot to behave in an appropriate manner over longer time scales, its behaviours have to be contingent and meaningful to the unfolding relationship. Recognising, interpreting and generating sustained and engaging social behaviours is as such an important—and essentially, open—research question. We believe that the recent progress of machine learning opens new opportunities in terms of both analysis and synthesis of complex social dynamics. To support these approaches, we introduce in this article a novel, open dataset of child social interactions, designed with data-driven research methodologies in mind. Our data acquisition methodology relies on an engaging, methodologically sound, but purposefully underspecified *free-play* interaction. By doing so, we capture a rich set of behavioural patterns occurring in natural social interactions between children. The resulting dataset, called the PInSoRo dataset, comprises 45+ hours of hand-coded recordings of social interactions between 45 child-child pairs and 30 child-robot pairs. In addition to annotations of social constructs, the dataset includes fully calibrated video recordings, 3D recordings of the faces, skeletal informations, full audio recordings, as well as game interactions.

Introduction

Studying social interactions

Studying social interactions requires a social *situation* that effectively elicits interactions between the participants. Such a situation is typically scaffolded by a social task, and consequently, the nature of this task influences in fundamental ways the kind of interactions that

available online: https://freeplay-sandbox.github.io/application.

Funding: This work was primarily funded by the European Union H2020 "Donating Robots a Theory of Mind" project (grant id #657227) awarded to SL. It received additional funding from the European Union H2020 "Second Language Tutoring using Social Robots" project (grant id #688014), awarded to TB. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

might be observed and analysed. In particular, the socio-cognitive tasks commonly found in both the experimental psychology and human-robot interaction (HRI) literature often have a narrow focus: because they aim at studying one (or a few) specific social or cognitive skills in isolation and in a controlled manner, these tasks are typically conceptually simple and highly constrained (for instance, object hand-over tasks; perspective-taking tasks; etc.). While these focused endeavours are important and necessary, they do not adequately reflect the complexity and dynamics of real-world, natural interactions (as discussed by Baxter et al. in [1], in the context of HRI). Consequently, we need to investigate richer interactions, scaffolded by socio-cognitive tasks that:

- are long enough and varied enough to elicit a large range of interaction situations;
- foster rich multi-modal interactions, such as simultaneous speech, gesture, and gaze behaviours;
- are not over-specified, in order to maximise natural, non-contrived behaviours;
- evidence complex social dynamics, such as rhythmic coupling, joint attention, implicit turn-taking;
- include a level of non-determinism and unpredictability.

The challenge lies in designing a social task that exhibits these features *while maintaining* essential scientific properties (repeatability; replicability; robust metrics) as well as good practical properties (not requiring unique or otherwise very costly experimental environments; not requiring very specific hardware or robotic platform; easy deployment; short enough experimental sessions to allow for large groups of participants).

Looking specifically at social interactions amongst children, we present in the next section our take on this challenge, and we introduce a novel task of free play. The task is designed to elicit rich, complex, varied social interactions while supporting rigorous scientific methodologies, and is well suited for studying both child-child and child-robot interactions.

Social play

Our interaction paradigm is based on free and playful interactions (hereafter, *free play*) in what we call a *sandboxed environment*. In other words, while the interaction is free (participants are not directed to perform any particular task beyond playing), the activity is both *scaffolded* and *constrained* by the setup mediating the interaction (a large interactive table), in a similar way to children freely playing with sand within the boundaries of a sandpit. Consequently, while participants engage in open-ended and non-directed activity, the play situation is framed to be easily reproducible as well as practical to record and analyse.

This initial description frames the socio-cognitive interactions that might be observed and studied: playful, dyadic, face-to-face interactions. While gestures and manipulations (including joint manipulations) play an important role in this paradigm, the participants do not typically move much during the interaction. Because it builds on play, this paradigm is also primarily targeted to practitioners in the field of child-child or child-robot social interactions.

The choice of a playful interaction is supported by the wealth of social situations and social behaviours that play elicits (see for instance parts 3 and 4 of [2]). Most of the research in this field builds on the early work of Parten who established five *stages of play* [3], corresponding to different stages of development, and accordingly associated with typical age ranges: (*a*) *solitary (independent) play* (age 2-3): child playing separately from others, with no reference to what others are doing; (*b*) *onlooker play* (age 2.5-3.5): child watching others play; may engage

in conversation but not engage in doing; true focus on the children at play; (*c*) *parallel play* (also called adjacent play, social co-action, age 2.5-3.5): children playing with similar objects, clearly beside others but not with them; (*d*) *associative play* (age 3-4): child playing with others without organization of play activity; initiating or responding to interaction with peers; (*e*) *cooperative play* (age 4+): coordinating one's behavior with that of a peer; everyone has a role, with the emergence of a sense of belonging to a group; beginning of "team work."

These five stages of play have been extensively discussed and refined over the last century, yet remain remarkably widely accepted. It must be noted that the age ranges are only indicative. In particular, most of the early behaviours still occur at times by older children.

Machine learning, robots and social behaviours

The data-driven study of social mechanisms is still an emerging field, and only limited literature is available.

The use of interaction datasets to teach artificial agents (robots) how to socially behave has been previously explored, and can be considered as the extension of the traditional learning from demonstration (LfD) paradigms to social interactions [4, 5]. However, existing research focuses on low-level identification or generation of brief, isolated behaviours, including social gestures [6] and gazing behaviours [7].

Based on a human-human interaction dataset, Liu et al. [8] have investigated machine learning approaches to learn longer interaction sequences. Using unsupervised learning, they train a robot to act as a shop-keeper, generating both speech and socially acceptable motions. Their approach remains task-specific, and they report only limited success. They however emphasise the "life-likeness" of the generated behaviours.

This burgeoning interest in the research community for the data-driven study of social responses is however impaired by the lack of structured research efforts. In particular, there is only limited availability of large and open datasets of social interactions, suitable for machine-learning applications.

One such dataset is the *Multimodal Dyadic Behavior Dataset* (*MMDB*, [9]). It comprises of 160 sessions of 3 to 5 minute child-adult interactions. During these interactions, the experimenter plays with toddlers (1.5 to 2.5 years old) in a semi-structured manner. The dataset includes video streams of the faces and the room, audio, physiological data (electrodermal activity) as well as manual annotations of specific behaviours (like gaze to the examiner, laughter, pointing). This dataset focuses on very young children during short, adult-driven interactions. As such, it does not include episodes of naturally-occurring social interactions between peers, and the diversity of said interactions is limited. Besides, the lack of intrinsic and extrinsic camera calibration information in the dataset prevent the automatic extraction and labeling of key interaction features (like mutual gaze).

Another recent dataset, the *Tower Game Dataset* [10], focuses specifically on rich dyadic social interactions. The dataset comprises of 39 adults recorded over a total of 112 annotated sessions of 3 min in average. The participants are instructed to jointly construct a tower using wooden blocks. Interestingly, the participants are not allowed to talk to maximise the amount of non-verbal communication. The skeletons and faces of the participants are recorded, and the dataset is manually annotated with so-called *Essential Social Interaction Predicates* (ESIPs): rhythmic coupling (entrainment or attunement), mimicry (behavioral matching), movement simultaneity, kinematic turn taking patterns, joint attention. This dataset does not appear to be publicly available on-line.

The UE-HRI dataset [11] is another recently published (2017) dataset of social interactions, focusing solely on human-robot interactions. 54 adult participants were recorded (duration

M = 7.7min) during spontaneous dialogues with a Pepper robot. The interactions took place in a public space, and include both one-to-one and multi-party interactions. The resulting dataset includes audio and video recordings from the robot perspective, as well as manual annotations of the levels of engagement. It is publicly available.

PInSoRo, our dataset, shares some of the aims of the *Tower Game* and *UE-HRI* datasets, with however significant differences. Contrary to these two datasets, our target population are children. We also put a strong focus on naturally occurring, real-world social behaviours. Furthermore, as presented in the following sections, we record much longer interactions (up to 40 minutes) of free play interactions, capturing a wider range of socio-cognitive behaviours. We did not place any constraints on the permissible communication modalities, and the recordings were manually annotated with a focus on social constructs.

Material and methods

The free-play sandbox task

As previously introduced, the *free-play sandbox* task is based on face-to-face free-play interactions, mediated by a large, horizontal touchscreen. Pairs of children (or alternatively, one child and one robot) are invited to freely draw and interact with items displayed on an interactive table, without any explicit goals set by the experimenter (Fig 1). The task is designed so that children can engage in open-ended and non-directive play. Yet, it is sufficiently constrained to

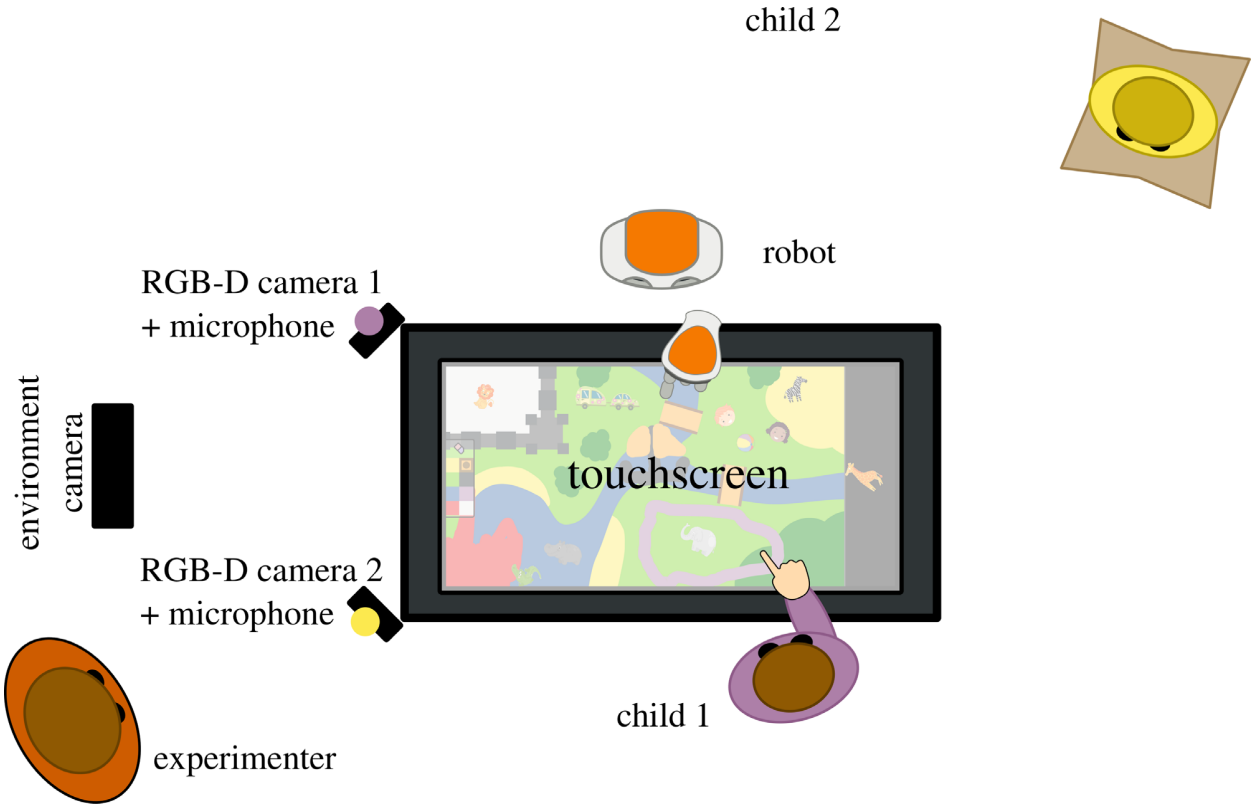

Fig 1. The free-play social interactions sandbox: Two children or one child and one robot (as pictured here) interacted in a free-play situation, by drawing and manipulating items on a touchscreen. Children were facing each other and sit on cushions. Each child wore a bright sports bib, either purple or yellow, to facilitate later identification.

https://doi.org/10.1371/journal.pone.0205999.g001

Fig 2. Example of a possible game situation. Game items (animals, characters. . .) can be dragged over the whole play area, while the background picture can be painted over by picking a colour. In this example, the top player is played by a robot.

https://doi.org/10.1371/journal.pone.0205999.g002

be suitable for recording, and allows the reproduction of social behaviour by an artificial agent in comparable conditions.

Specifically, the free-play sandbox follows the *sandtray* paradigm [12]: a large touchscreen (60cm × 33cm, with multitouch support) is used as an interactive surface. The two players, facing each other, play together, moving interactive items or drawing on the surface if they wish so (Fig 2). The background image depicts a generic empty environment, with different symbolic colours (water, grass, beach, bushes. . .). By drawing on top of the background picture, the children can change the environment to their liking. The players do not have any particular task to complete, they are simply invited to freely play. They can play for as long as they wish. However, for practical reasons, we had to limit the sessions to a maximum of 40 minutes.

Even though the children do typically move a little, the task is fundamentally a face-to-face, spatially delimited, interaction, and as such simplifies the data collection. In fact, the children's faces were successfully detected in 98% of the over 2 million frames recorded during the PInSoRo dataset acquisition campaign.

Experimental conditions. The PInSoRo dataset aims to establish two experimental baselines for the free-play sandbox task: the 'human social interactions' baseline on one hand (child–child condition), an 'asocial' baseline on the other hand (child–*non-social* robot condition). These two baselines aim to characterise the qualitative and quantitative bounds of the spectrum of social interactions and dynamics that can be observed in this situation.

In the *child-child* condition, a diverse set of social interactions and social dynamics were expected to be observed, ranging from little social interactions (for instance, with shy children) to strong, positive interactions (for instance, good friends), to hostility (children who do not get along very well).

In the *asocial* condition, one child was replaced by an autonomous robot. The robot was purposefully programmed to be *asocial*. It autonomously played with the game items as a child would (although it did not perform any drawing action), but avoided all social interactions: no social gaze, no verbal interaction, no reaction to child-initiated game actions.

From the perspective of social psychology, this condition provides a baseline for the social interactions and dynamics at play (or the lack thereof) when the social communication channel is severed between the agents, while maintaining a similar social setting (face-to-face interaction; free-play activity).

From the perspective of human-robot interaction and artificial intelligence in general, the child–'asocial robot' condition provides a baseline to contrast with for yet-to-be-created richer social and behavioural AI policies.

Hardware apparatus. The interactive table was based on a 27" Samsung All-In-One computer (quad core i7-3770T, 8GB RAM) running Ubuntu Linux and equipped with a fast 1TB SSD hard-drive. The computer was held horizontally in a custom aluminium frame standing 26cm above the floor. All the cameras were connected to the computer via USB-3. The computer performed all the data acquisition using ROS Kinetic (http://www.ros.org/). The same computer was also running the game interface on its touch-enabled screen (60cm × 33cm), making the whole system standalone and easy to deploy.

The children's faces were recorded using two short range (0.2m to 1.2m) Intel RealSense SR300 RGB-D cameras placed at the corners of the touchscreen (Fig 1) and tilted to face the children. The cameras were rigidly mounted on custom 3D-printed brackets. This enabled a precise measurement of their 6D pose relative to the touchscreen (extrinsic calibration).

Audio was recorded from the same SR300 cameras (one mono audio stream was recorded for each child, from the camera facing him or her).

Finally, a third RGB camera (the RGB stream of a Microsoft Kinect One, the *environment camera* in Fig 1) recorded the whole interaction setting. This third video stream was intended to support human coders while annotating the interaction, and was not precisely calibrated.

In the child-robot condition, a Softbank Robotics' Nao robot was used. The robot remained in standing position during the entire play interaction. The actual starting position of the robot with respect to the interactive table was recalibrated before each session by flashing a 2D fiducial marker on the touchscreen, from which the robot could compute its physical location.

Software apparatus. The software-side of the free-play sandbox is entirely open-source (source code: https://github.com/freeplay-sandbox/). It was implemented using two main frameworks: Qt QML (http://doc.qt.io/qt-5/qtquick-index.html) for the user interface (UI) of the game (Fig 2), and the *Robot Operating System* (ROS) for the modular implementation of the data processing and behaviour generation pipelines, as well as for the recordings of the various datastreams (Fig 4). The graphical interface interacts with the decisional pipeline over a bidirectional QML-ROS bridge that was developed for that purpose (source code available from the same link).

Fig 3 presents the complete software architecture of the sandbox as used in the child-robot condition (in the child-child condition, robot-related modules were simply not started).

Robot control. As previously described, one child was replaced by a robot in the child-robot condition. Our software stack allowed for the robot to be used in two modes of operations: either autonomous (selecting actions based on pre-programmed play policies), or controlled by a human operator (so-called *Wizard-of-Oz* mode of operation).

For the purpose of the PInSoRo dataset, the robot behaviour was fully autonomous, yet coded to be purposefully *asocial* (no social gaze, no verbal interaction, no reaction to child-initiated game actions). The simple action policy that we implemented consisted in the robot choosing a random game item (in its reach), and moving that item to a predefined zone on the

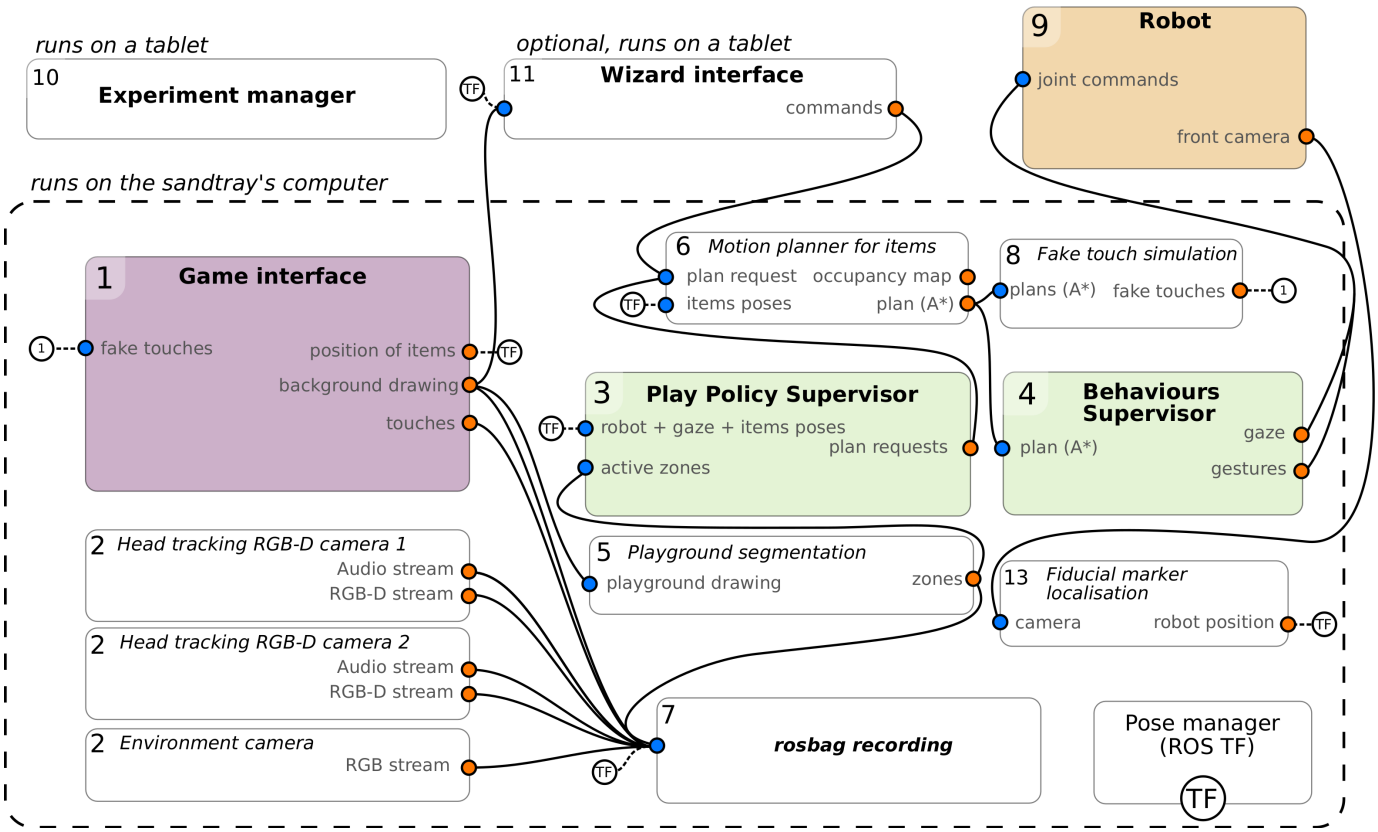

Fig 3. Software architecture of the free-play sandbox (data flows *from* orange dots *to* blue dots). Left nodes interact with the interactive table hardware (game interface (1) and camera drivers (2)). The green nodes in the centre implement the behaviour of the robot (play policy (3) and robot behaviours (4)). Several helper nodes are available to provide for instance a segmentation of the children drawings into zones (5) or A* motion planning for the robot to move in-game items (6). Nodes are implemented in Python (except for the game interface, developed in QML) and inter-process communication relies on ROS. 6D poses are managed and exchanged via ROS TF.

https://doi.org/10.1371/journal.pone.0205999.g003

map (e.g. if the robot could reach the crocodile figure, it would attempt to drag it to a blue, i.e. water, zone). The robot did not physically drag the item on the touchscreen: it relied on a A* motion planner to find an adequate path, sent the resulting path to the touchscreen GUI to animate the displacement of the item, and moved its arm in a synchronized fashion using the inverse kinematics solver provided with the robot's software development kit (SDK).

In the Wizard-of-Oz mode of operation, the experimenter would remotely control the robot through a tablet application developed for this purpose (Figs 3–11). The tablet exactly mirrored the game state, and the experimenter dragged the game items on the tablet as would the child on the touchscreen. On release, the robot would again mimic the dragging motion on the touchscreen, moving an object to a new location. This mode of operation, while useful to conduct controlled studies, was not used for the dataset acquisition.

Experiment manager. We developed as well a dedicated web-based interface (usually accessed from a tablet) for the experimenter to manage the whole experiment and data acquisition procedure (Figs 3–10). This interface ensured that all the required software modules were running; it allowed the experimenter to check the status of each of them and, if needed, to start/stop/restart any of them. It also helped managing the data collection campaign by

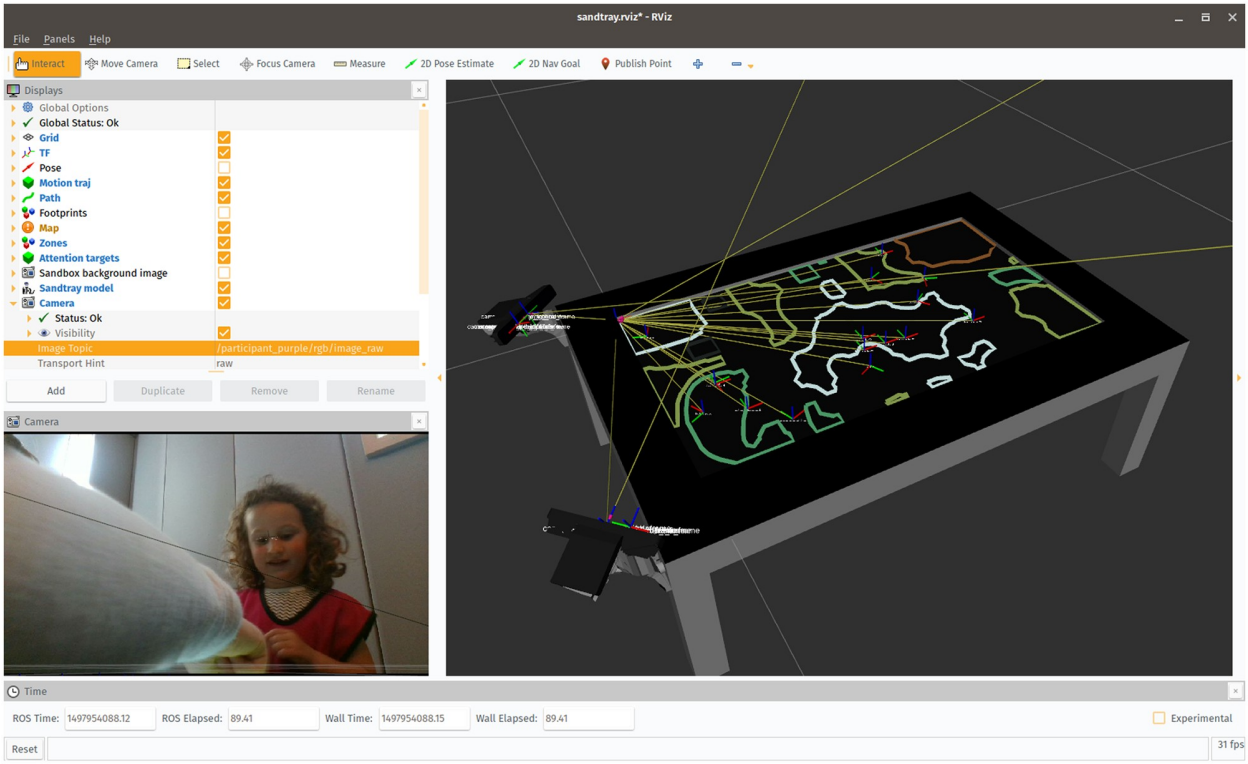

Fig 4. The free-play sandbox, viewed at runtime within ROS RViz. Simple computer vision was used to segment the background drawings into zones (visible on the right panel). The poses and bounding boxes of the interactive items were broadcast as well, and turned into an occupancy map, used to plan the robot's arm motion. The individual pictured in this figure has given written informed consent (as outlined in PLOS consent form) to appear.

https://doi.org/10.1371/journal.pone.0205999.g004

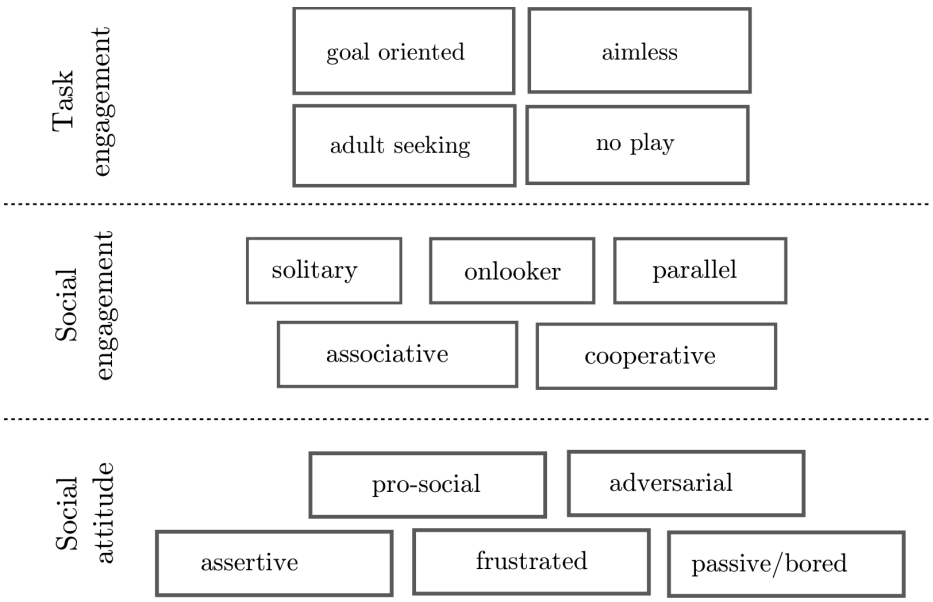

Fig 5. The coding scheme used for annotating social interactions occurring during free-play episodes. Three main axis were studied: task engagement, social engagement and social attitude.

https://doi.org/10.1371/journal.pone.0205999.g005

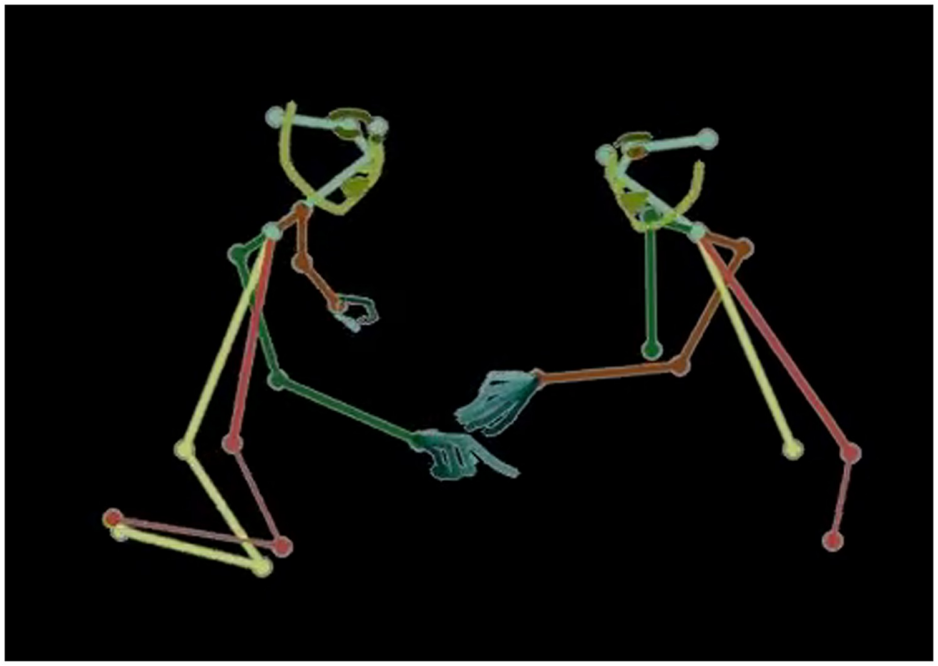

Fig 6. 2D skeletons, including facial landmarks and hand details are automatically extracted using the OpenPose library [18].

https://doi.org/10.1371/journal.pone.0205999.g006

providing a convenient interface to record the participants' demographics, resetting the game interface after each session, and automatically enforcing the acquisition protocol (presented in Table 1).

Coding of the social interactions

Our aim is to provide insights on the social dynamics, and as such we annotated the dataset using a combination of three coding schemes for social interactions that reuse and adapt established social scales. Our resulting coding scheme (Fig 5) looked specifically at three axis: the level of *task engagement* (that distinguishes between *focused*, *task oriented* behaviours, and *disengaged*—yet sometimes highly social – behaviours); the level of social engagement (reusing Parten's stages of play, but at a fine temporal granularity); the social attitude (that encoded attitudes like *supportive*, *aggressive*, *dominant*, *annoyed*, etc).

Task engagement. The first axis of our coding scheme aimed at making a broad distinction between 'on-task' behaviours (even though the free-play sandbox did not explicitly require the children to perform a specific task, they were still engaged in an underlying task: to play with the game) and 'off-task' behaviours. We called 'on-task' behaviours *goal oriented*: they encompassed considered, planned actions (that might be social or not). *Aimless* behaviours (with respect to the task) encompassed opposite behaviours: being silly, chatting about unrelated matters, having a good laugh, etc. These *Aimless* behaviours were in fact often highly social, and played an important role in establishing trust and cooperation between the peers. In that sense, we considered them as as important as on-task behaviours.

Social engagement: Parten's stages of play at micro-level. In our scheme, we characterised *Social engagement* by building upon Parten's stages of play [3]. These five stages of play

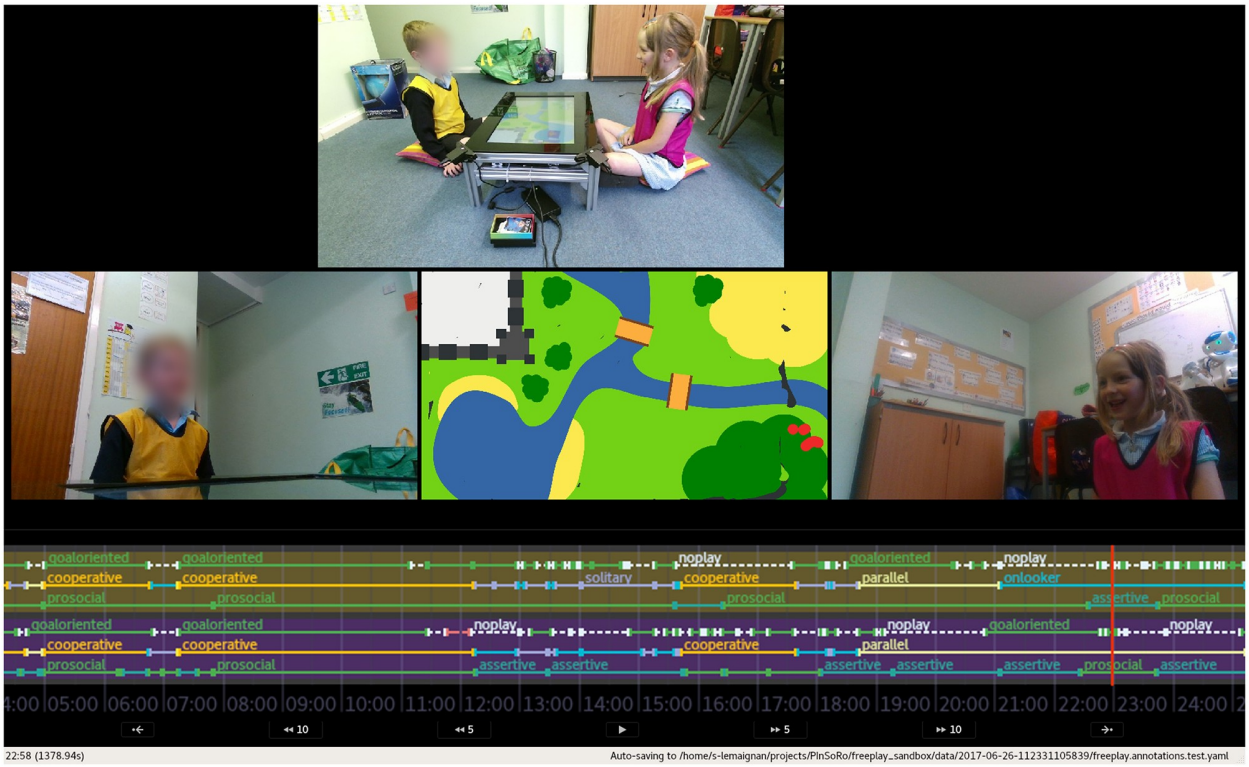

Fig 7. Screenshot of the dedicated tool developed for rapid annotation of the social interactions. The annotators used a secondary screen (tablet) with buttons (layout similar to Fig 5) to record the social constructs. Figure edited for legibility (timeline enlarged) and to mask out one of the children' face. The right individual pictured in this figure has given written informed consent (as outlined in PLOS consent form) to appear.

https://doi.org/10.1371/journal.pone.0205999.g007

are normally used to characterise rather long sequences (at least several minutes) of social interactions. In our coding scheme, we applied them at the level of each of the micro-sequences of the interactions: one child is drawing and the other is observing was labelled as *solitary play* for the former child, *on-looker* behaviour for the later; the two children discuss what to do next: this sequence was annotated as a *cooperative* behaviour; etc.

We chose this fine-grained coding of social engagement to enable proper analyses of the internal dynamics of a long sequence of social interaction.

Social attitude. The constructs related to the social *attitude* of the children derived from the *Social Communication Coding System* (SCCS) proposed by Olswang et al. [13]. The SCCS consists in 6 mutually exclusive constructs characterising social communication (*hostile*; *pro-social*; *assertive*; *passive*; *adult seeking*; *irrelevant*) and were specifically created to characterise children's communication in a classroom setting.

We transposed these constructs from the communication domain to the general behavioural domain, keeping the *pro-social*, *hostile* (whose scope we broadened in *adversarial*), *assertive* (i.e. dominant), and *passive* constructs. In our scheme, the *adult seeking* and *irrelevant* constructs belong to Task Engagement axis.

Finally, we added the construct *Frustrated* to describe children who are reluctant or refuse to engage in a specific phase of interaction because of a perceived lack of fairness or attention from their peer, or because they fail at achieving a particular task (like a drawing).

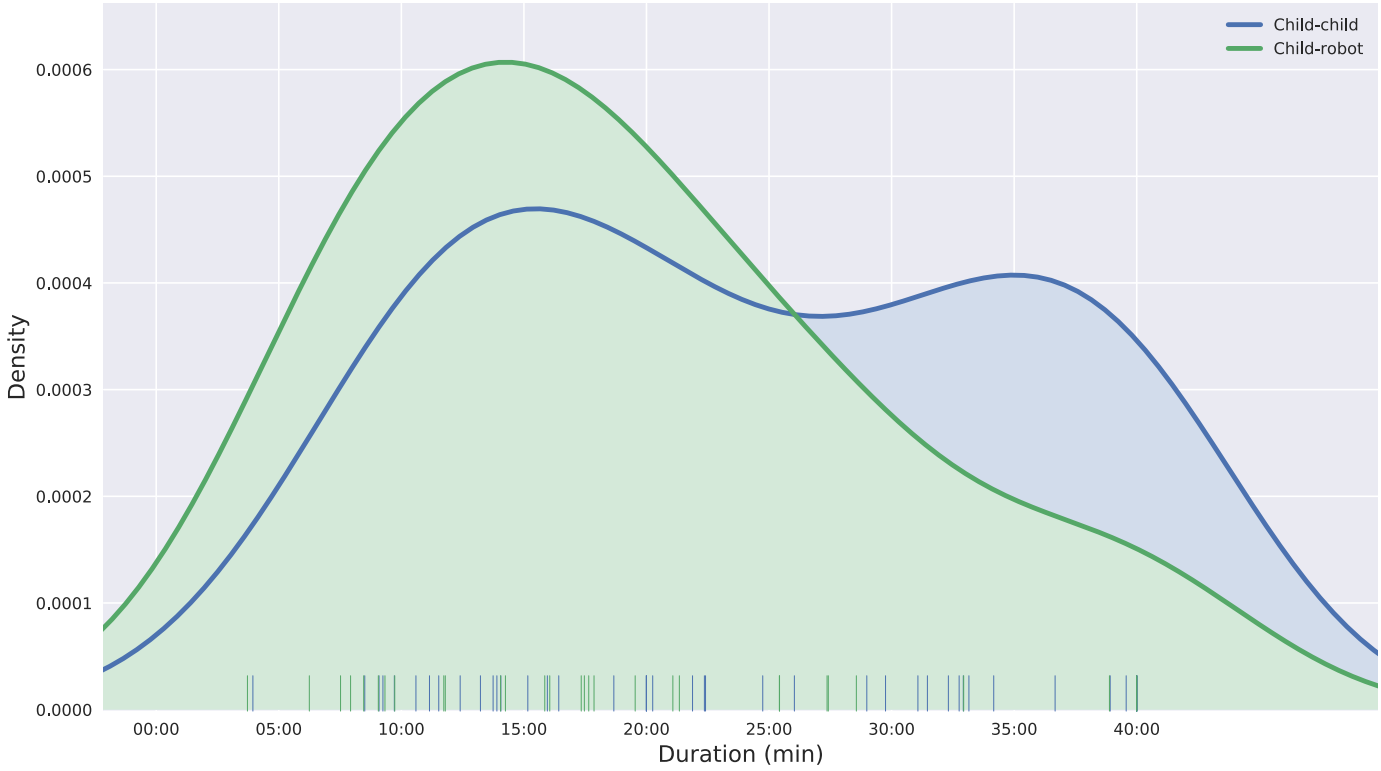

Fig 8. Density distribution of the durations of the interactions for the two conditions. Interactions in the child-robot condition were generally shorter than the child-child interactions. Interactions in the child-child condition followed a bi-modal distribution, with one mode centered around minute 15 (similar to the child-robot one) and one, much longer mode, at minute 37.

https://doi.org/10.1371/journal.pone.0205999.g008

Protocol

We adhered to the acquisition protocol described in Table 1 with all participants. To ease later identification, each child was also given a different and brightly coloured sports bib to wear.

Importantly, during the *Greetings* stage, we showed the robot both moving and speaking (for instance, "Hello, I'm Nao. Today I'll be playing with you. Exciting!" while waving at the children). This was of particular importance in the child-robot condition, as it set the children's expectations in term of the capabilities of the robot: the robot could in principle speak, move, and even behave in a social way.

Also, the game interface of the free-play sandbox offered a tutorial mode, used to ensure the children know how to manipulate items on a touchscreen and draw. In our experience, this never was an issue for children.

Data collection

Table 2 lists the raw datastreams that were collected during the game. By relying on ROS for the data acquisition (and in particular the `rosbag` tool), we ensured all the datastreams were synchronised, timestamped, and, where appropriate, came with calibration information (for the cameras mainly). For the PInSoRo dataset, cameras were configured to stream in qHD resolution (960×540 pixels) in an attempt to balance high enough resolution with tractable file size. It resulted in bag files weighting ≈1GB per minute.

Besides audio and video streams, user interactions with the game were monitored and recorded as well. The background drawings produced by the children were recorded. They

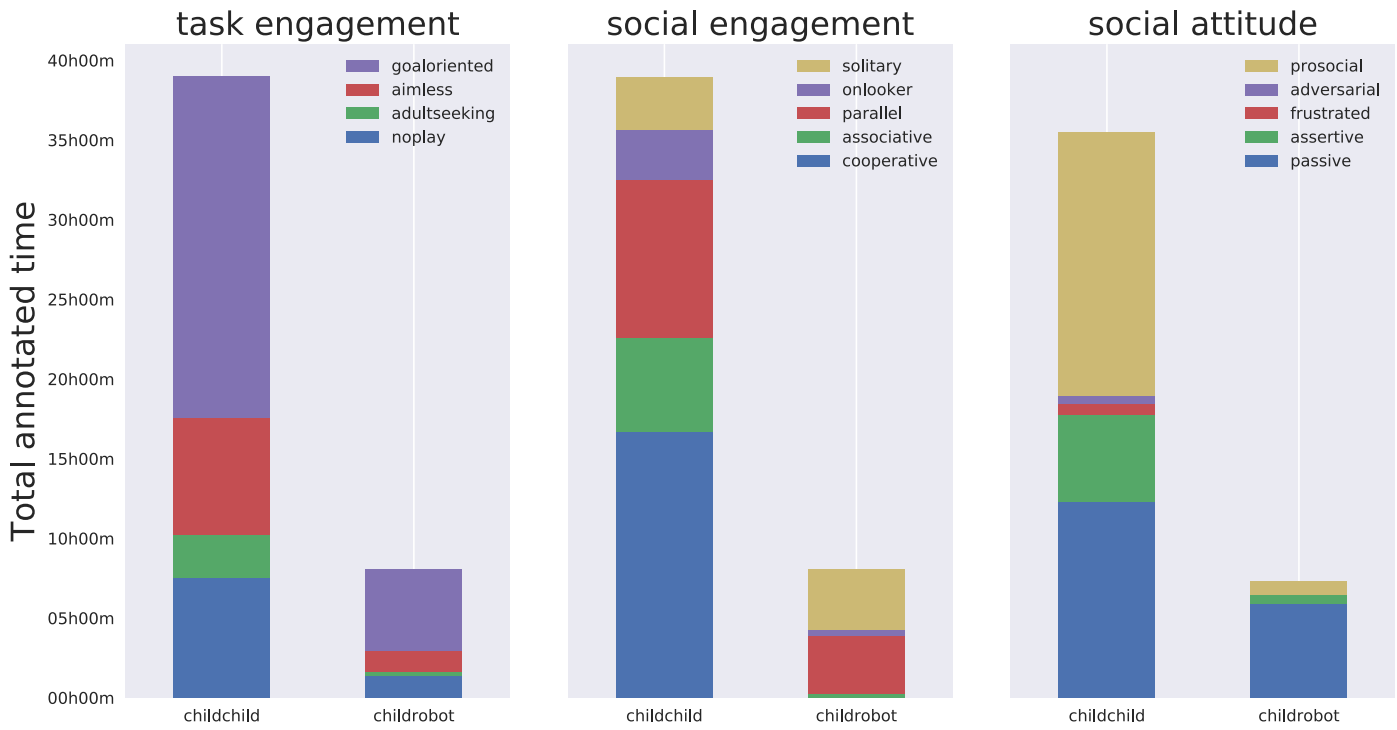

Fig 9. Repartition of annotations over the dataset (in total duration of recordings annotated with a given construct). The three classes of constructs (task engagement, social engagement, social attitude) and the two conditions (child-child and child-robot) are plotted separately.

https://doi.org/10.1371/journal.pone.0205999.g009

were also segmented according to their colours, and the contours of resulting regions were extracted and recorded. The positions of all manipulable game items were recorded (as ROS TF frames), as well as every touch on the touchscreen.

Data post-processing

Table 3 summarises the post-processed datastreams that are made available alongside the raw datastreams.

Audio processing. Audio features were automatically extracted using the OpenSMILE toolkit [14]. We used a 33ms-wide time windows in order to match the cameras FPS. We extracted the INTERSPEECH 2009 Emotion Challenge standardised features [15]. These are a range of prosodic, spectral and voice quality features that are arguably the most common features we might want to use for emotion recognition [16]. For a full list, please see [15]. As no reliable speech recognition engine for children voice could be found [17], audio recordings were not automatically transcribed.

Facial landmarks, action-units, skeletons, gaze. Offline post-processing was performed on the images obtained from the cameras. We relied on the CMU OpenPose library [18] to extract for each child the upper-body skeleton (18 points), 70 facial landmarks including the pupil position, as well as the hands' skeleton (Fig 6).

This skeletal information was extracted from the RGB streams of each of the three cameras, for every frame. It is stored alongside the main data in an easy-to-parse JSON file.

For each frame, 17 action units, with accompanying confidence levels, were also extracted using the OpenFace library [19]. The action-units recognised by OpenFace and provided

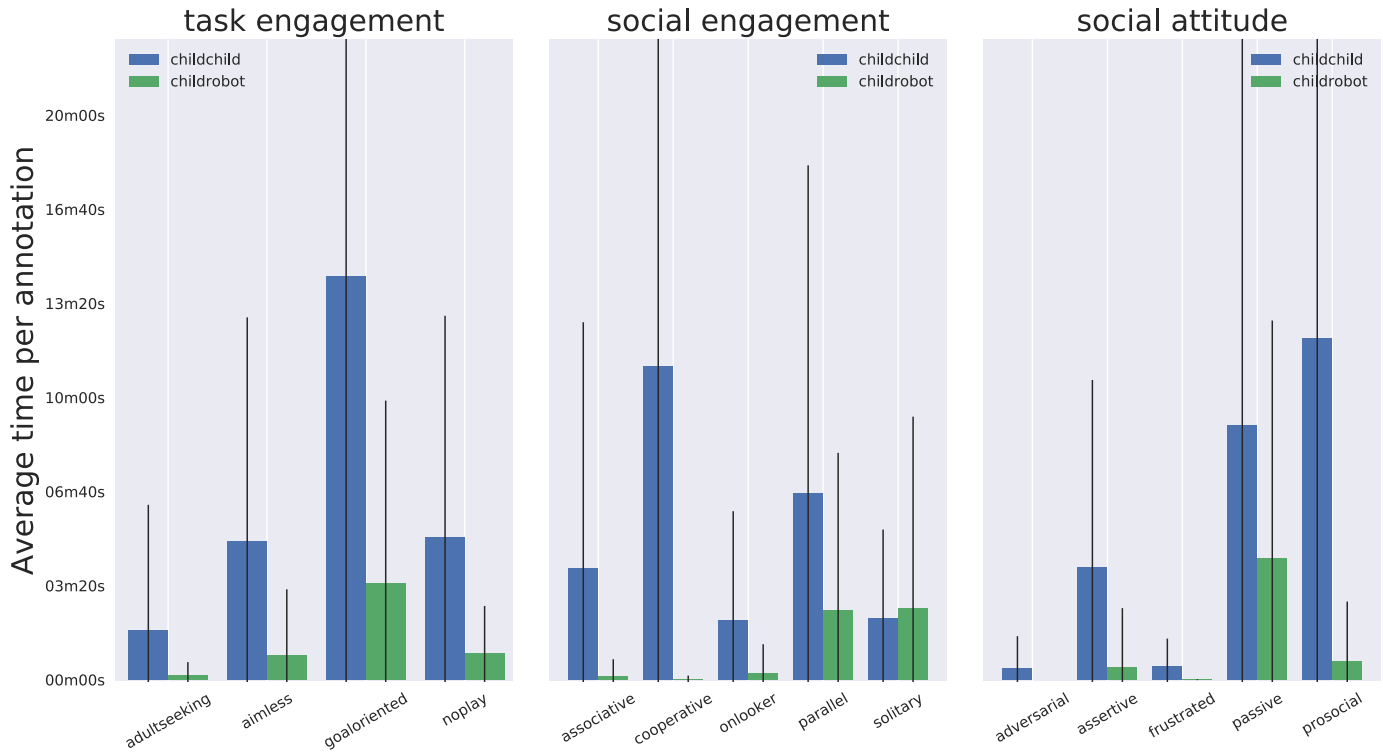

Fig 10. Mean time (and standard deviation) that each construct has been annotated in each recording. The large standard deviations reflect the broad range of group dynamics captured in the dataset.

https://doi.org/10.1371/journal.pone.0205999.g010

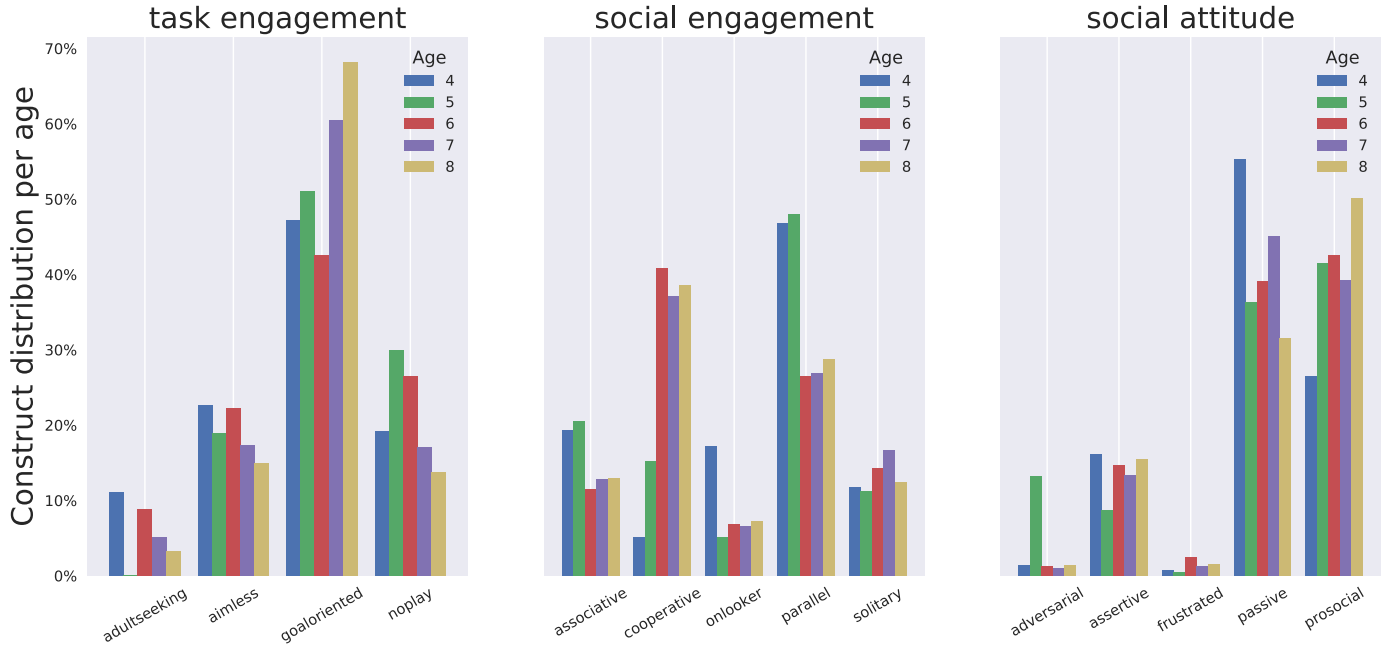

Fig 11. Percentage of observations for each constructs with respect the children's age.

https://doi.org/10.1371/journal.pone.0205999.g011

Table 1. Data acquisition protocol.

Greetings *(about 5 min)* • explain the purpose of the study: showing robots how children play • briefly present a Nao robot: the robot stands up, gives a short message (*Today I'll be watching you playing* in the child-child condition;*Today I'll be playing with you* in the child-robot condition), and sits down. • place children on cushions • complete demographics on the tablet • remind the children that they can withdraw at anytime
Gaze tracking task *(40 sec)* children are instructed to closely watch a small picture of a rocket that moves randomly on the screen. Recorded data is used to train a eye-tracker post-hoc.
Tutorial *(1-2 min)* explain how to interact with the game, ensure the children are confident with the manipulation/drawing.
Free-play task (up to 40 min) • initial prompt: *"Just to remind you, you can use the animals or draw. Whatever you like. If you run out of ideas, there's also an ideas box. For example, the first one is a zoo. You could draw a zoo or tell a story. When you get bored or don't want to play anymore, just let me know."* • let children play • once they wish to stop, stop recording
Debriefing *(about 2 min)* • answer possible questions from the children • give small reward (e.g. stickers) as a thank you

https://doi.org/10.1371/journal.pone.0205999.t001

Table 2. List of raw datastreams available in the PInSoRo dataset. Each datastream is timestamped with a synchronised clock to facilitate later analysis.

Domain	Type	Details
child 1	audio	16kHz, mono, semi-directional
	face (RGB)	qHD (960×540), 30Hz
	face (depth)	VGA (640×480), 30Hz
child 2	audio	16kHz, mono, semi-directional
	face (RGB)	qHD (960×540), 30Hz
	face (depth)	VGA (640×480), 30Hz
environment	RGB	qHD (960×540), 29.7Hz
game interactions	background drawing (RGB)	4Hz
	finger touches	6 points multi-touch, 10Hz
	game items pose	TF frames, 10Hz
other	static transforms between touchscreen and facial cameras	
	cameras calibration informations	

https://doi.org/10.1371/journal.pone.0205999.t002

alongside the data are AU01, AU02, AU04, AU05, AU06, AU07, AU09, AU10, AU12, AU14, AU15, AU17, AU20, AU23, AU25, AU26, AU28 and AU45 (classification following https://www.cs.cmu.edu/~face/facs.htm).

Gaze was also estimated, using two techniques. First, head pose estimation was performed following [20], and used to estimate gaze pose. While this technique is effective to segment pose at a coarse level (i.e. gaze on interactive table vs. gaze on other child/robot vs. gaze on experimenter), it offers limited accuracy when tracking the precise gaze location on the surface of the interactive table (due to not tracking the eye pupils).

We complemented head pose estimation with a neural network (a simple 7-layers, fully connected, multi-layer perceptron with ReLU activations and 64 units per layer), implemented

Table 3. List of post-processed datastreams available in the PInSoRo dataset. With the exception of social annotations, all the data was automatically computed from the raw datastreams at 30Hz.

Domain	Type	Details
children	face	70 facial landmarks (2D)
		17 facial action-units
		head pose estimation (TF frame)
		gaze estimation (TF frame)
	skeleton	18 points body pose (2D)
		20 points hand tracking (2D, only when visible)
	audio	INTERSPEECH's 16 low-level descriptors
annotations	timestamped annotations of social behaviours and remarkable events	

https://doi.org/10.1371/journal.pone.0205999.t003

with the Caffe framework (source available here: https://github.com/severin-lemaignan/visual_tracking_caffe).

The network trained from a ground truth mapping between the children' faces and 2D gaze coordinates. Training data is obtained by asking the children to follow a target on the screen for a short period of time before starting the main free play activity (see protocol, Table 1). The position of the target provides the ground truth (x, y) coordinates of the gaze on the screen. For each frame, the network is then fed a feature vector comprising 32 facial and skeletal (x, y) points of interest relevant to gaze estimation (namely, the 2D location of the pupils, eye contours, eyebrows, nose, neck, shoulders and ears). The training dataset comprises 80% of the fully randomized dataset (123711 frames) and the testing dataset the remaining 20% (30927 frames). Using this technique, we measured a gaze location error of 12.8% on our test data between the ground truth location of the target on the screen and the estimated gaze location (i.e. ±9cm over the 70cm-wide touchscreen). The same pre-trained network is then used to provide gaze estimation during the remainder of the free play activity.

Video coding. The coding was performed post-hoc with the help of a dedicated annotation tool (Fig 7) which is part of the free-play sandbox toolbox. This tool can replay and randomly seek in the three video streams, synchronised with the recorded state of the game (including the drawings as they were created). An interactive timeline displaying the annotations is also displayed.

The annotation tool offers a remote interface for the annotator (made of large buttons, and visually similar to Fig 5) that is typically displayed on a tablet and allow the simultaneous coding of the behaviours of the two children. Usual video coding practices (double-coding of a portion of the dataset and calculation of an inter-judge agreement score) were followed.

Results—The PInSoRo dataset

Using the free-play sandbox methodology, we have acquired a large dataset of social interactions between either pairs of children or one child and one robot. The data collection took place over a period of 3 months during Spring 2017.

In total, 120 children were recorded for a total duration of 45 hours and 48 minutes of data collection. These 120 children (see demographics in Table 4; sample drawn from local schools) were randomly assigned to one of two conditions: the child-child condition (90 children, 45 pairs) and a child-robot condition (30 children). The sample sizes were balanced in favour of the child-child condition as the social dynamics that we ultimately want to capture are much richer in this condition.

Table 4. Descriptive statistics for the children.

Condition	Age Mean	Age SD	# girls	# boys
Whole group	6.4	1.3	55	65
Child-child	6.3	1.4	42	48
Child-robot	6.9	0.9	12	18

https://doi.org/10.1371/journal.pone.0205999.t004

In both conditions, and after a short tutorial, the children were simply invited to freely play with the sandbox, for as long as they wished (with a cap at 40 min; cf. protocol in Table 1).

In the child-child condition, 45 free-play interactions (i.e. 90 children) were recorded with a mean duration M = 24.15 min (standard deviation SD = 11.25 min). In the child-robot condition, 30 children were recorded, M = 19.18 min (SD = 10 min).

Fig 8 presents the density distributions of the durations of the interactions for the two baselines. The distributions show that (1) the vast majority of children engaged easily and for non-trivial amounts of time with the task; (2) the task led to a wide range of levels of commitment, which is desirable: it supports the claim that the free-play sandbox is an effective paradigm to observe a range of different social behaviours; (3) many long interactions (>30 min) were observed, which is especially desirable to study social dynamics.

The distribution of the child-robot interaction durations shows that these interactions are generally shorter. This was expected as the robot's asocial behaviour was designed to be less engaging. Often, the child and the robot were found to be playing side-by-side—in some case for rather long periods of time—without interacting at all (solitary play).

Over the whole dataset, the children faces were detected on 98% of the images, which validates the positioning of the camera with respect to the children to record facial features.

Annotations

Five expert annotators performed the dataset annotation. Each annotator received one hour of training by the experimenters, and were compensated for their work.

In total, 13289 annotations of social dynamics were produced, resulting in an average of 149 annotations per record (SD = 136), which equates to an average of 4.2 annotations/min (SD = 2.1), and an average duration of annotated episodes of 48.8 sec (SD = 33.3). Fig 9 shows the repartition of the annotation corpus over the different constructs presented in Fig 5. Fig 10 shows the mean annotation time and standard deviation per recording for each construct.

Overall, 23% of the dataset was double-coded. Inter-coder agreement was found to be 51.8% (SD = 16.8) for task engagement annotations; 46.1% (SD = 24.2) for social engagement; 56.6% (SD = 22.9) for social attitude.

These values are relatively low (only partial agreement amongst coders). This was expected, as annotating social interactions beyond surface behaviours is indeed generally difficult. The observable, objective behaviours are typically the result of a superposition of the complex and non-observable underlying cognitive and emotional states. As such, these deeper socio-cognitive states can only be indirectly observed, and their labelling is typically error prone.

However, this is not anticipated to be a major issue for data-driven analyses, as machine learning algorithms are typically trained to estimate probability distributions. As such, divergences in human interpretations of a given social episode will simply be reflected in the probability distribution of the learnt model.

When looking at social behaviours with respect to age groups, expected behavioural trends are observed (Fig 11): *adult seeking* goes down when children get older; more *cooperative* play

is observed with older children, while more *parallel* play takes place with younger ones. In constrast, the social attitudes appear evenly distributed amongst age groups.

Dataset availability and data protection

All data has been collected by researchers at the University of Plymouth, under a protocol approved by the university ethics committee. The parents of the participants explicitly consented in writing to sharing of their child's video and audio with the research community. The data does not contain any identifying information, except the participant's images. The child's age and gender are also available. The parents of the children in this manuscript have given written informed consent (as outlined in PLOS consent form) to publish these case details.

The dataset is freely available to any interested researcher. Due to ethical and data protection regulations, the dataset is however made available in two forms: a public, Creative Commons licensed, version that does not include any video material of the children (no video nor audio streams), and hosted on the Zenodo open-data platform: https://zenodo.org/record/1043508. The complete version that includes all video streams is freely available as well, but interested researchers must first fill a data protection form. The detail of the procedure are available online: https://freeplay-sandbox.github.io/application.

Discussion of the free-play sandbox

The free-play sandbox elicits a loosely structured form of play: the actual play situations are not known beforehand and might change several times during the interaction; the game actions, even though based on one primary interaction modality (touches on the interactive table), are varied and unlimited (especially when considering the drawings); the social interactions between participants are multi-modal (speech, body postures, gestures, facial expressions, etc.) and unconstrained. This loose structure creates a fecund environment for children to express a range of complex, dynamics, natural social behaviours that are not tied to an overly constructed social situation. The diversity of the social behaviours that we have been able to capture can indeed been seen in Figs 9 and 11.

Yet, the interaction is nonetheless structured. First, the physical bounds of the interactive table limit the play area to a well defined and relatively small area. As a consequence, children are mostly static (they are sitting in front of the table) and their primary form of physical interaction is based on 2D manipulations on a screen.

Second, the game items themselves (visible in Fig 2) structure the game scenarios. They are iconic characters (animals or children) with strong semantics associated to them (such as 'crocodiles like water and eat children'). The game background, with its recognizable zones, also elicit a particular type of games (like building a zoo or pretending to explore the savannah).

These elements of structure (along with other, like the children demographics) arguably limit how general the PInSoRo dataset is. However, it also enable the free-play sandbox paradigm to retain key properties that makes it a practical and effective scientific tool: because the game builds on simple and universal play mechanics (drawings, pretend play with characters), the paradigm is essentially cross-cultural; because the sandbox is physically bounded and relatively small, it can be easily transported and practically deployed in a range of environments (schools, exhibitions, etc.); because the whole apparatus is well defined and relatively easy to duplicate (it essentially consists in one single touchscreen computer), the free-play sandbox facilitates the replication of studies while preserving ecological validity.

Compared to existing datasets of social interactions (the *Multimodal Dyadic Behavior Dataset*, the *Tower Game* dataset and the *UE-HRI* dataset), PInSoRo is much larger, with more than

45 hours of data, compared to 10.6, 5.6 and 6.9 hours respectively. PInSoRo is fully multimodal whereas the *Tower Game* dataset does not include verbal interactions, and the *UE-HRI* dataset focuses instead of spoken interactions. Compared to the *Multimodal Dyadic Behavior Dataset*, PInSoRo captures a broader range of social situations, with fully calibrated datastreams, enabling a broad range of automated data processing and machine learning applications. Finally, PInSoRo is also unique for being the first (open) dataset capturing *long sequences* (up to 40 minutes) of *ecologically valid* social interactions amongst children or between children and robots.

Conclusion—Towards the machine learning of social interactions?

We presented in this article the PInSoRo dataset, a large and open dataset of loosely constrained social interactions between children and robots. By relying on prolonged free-play episodes, we captured a rich set of naturally-occurring social interactions taking place between pairs of children or pairs of children and robots. We recorded an extensive set of calibrated and synchronised multimodal datastreams which can be used to mine and analyse the social behaviours of children. As such, this data provides a novel playground for the data-driven investigation and modelling of the social and developmental psychology of children.

The PInSoRo dataset also holds considerable promise for the automatic training of models of social behaviours, including implicit social dynamics (like rhythmic coupling, turn-taking), social attitudes, or engagement interpretation. As such, we foresee that the dataset might play an instrumental role in enabling artificial systems (and in particular, social robots) to recognise, interpret, and possibly, generate, socially congruent signals and behaviours whenever interacting with children. Whether such models can help uncover some of the implicit precursors of social behaviours, and is so, whether the same models, learnt from children data, can as well be used to interpret adult social behaviours, are open—and stimulating—questions that this dataset might contribute to answer.

Acknowledgments

The authors warmly thank the Plymouth's BabyLab, Freshlings nursery, Mount Street Primary School and Salisbury Road Primary School for their help with data acquisition. We also want to gratefully acknowledge the annotation work done by Lisa, Scott, Zoe, Rebecca and Sally.

This work has been supported by the EU H2020 Marie Sklodowska-Curie Actions project DoRoThy (grant 657227) and the H2020 L2TOR project (grant 688014).

Author Contributions

Conceptualization: Séverin Lemaignan, Tony Belpaeme.

Data curation: Charlotte E. R. Edmunds.

Formal analysis: Charlotte E. R. Edmunds.

Funding acquisition: Tony Belpaeme.

Investigation: Séverin Lemaignan, Charlotte E. R. Edmunds.

Methodology: Séverin Lemaignan, Charlotte E. R. Edmunds.

Software: Séverin Lemaignan, Emmanuel Senft.

Supervision: Séverin Lemaignan, Tony Belpaeme.

Writing – original draft: Séverin Lemaignan.

Writing – review & editing: Séverin Lemaignan, Charlotte E. R. Edmunds, Emmanuel Senft.

References

1. Baxter P, Kennedy J, E S, Lemaignan S, Belpaeme T. From Characterising Three Years of HRI to Methodology and Reporting Recommendations. In: Proceedings of the 2016 ACM/IEEE Human-Robot Interaction Conference (alt.HRI); 2016.
2. Bruner JS, Jolly A, Sylva K, editors. Play: Its role in development and evolution. Penguin; 1976.
3. Parten MB. Social participation among pre-school children. The Journal of Abnormal and Social Psychology. 1932; 27(3):243. https://doi.org/10.1037/h0074524
4. Nehaniv CL, Dautenhahn K. Imitation and social learning in robots, humans and animals: behavioural, social and communicative dimensions. Cambridge University Press; 2007.
5. Mohammad Y, Nishida T. Interaction Learning Through Imitation. In: Data Mining for Social Robotics. Springer; 2015. p. 255–273.
6. Nagai Y. Learning to comprehend deictic gestures in robots and human infants. In: Proc. of the 14th IEEE Int. Symp. on Robot and Human Interactive Communication. IEEE; 2005. p. 217–222.
7. Calinon S, Billard A. Teaching a humanoid robot to recognize and reproduce social cues. In: Proc. of the 15th IEEE Int. Symp. on Robot and Human Interactive Communication. IEEE; 2006. p. 346–351.
8. Liu P, Glas DF, Kanda T, Ishiguro H, Hagita N. How to Train Your Robot—Teaching service robots to reproduce human social behavior. In: Proc. of the 23rd IEEE Int. Symp. on Robot and Human Interactive Communication; 2014. p. 961–968.
9. Rehg J, Abowd G, Rozga A, Romero M, Clements M, Sclaroff S, et al. Decoding children's social behavior. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2013. p. 3414–3421.
10. Salter DA, Tamrakar A, Siddiquie B, Amer MR, Divakaran A, Lande B, et al. The tower game dataset: A multimodal dataset for analyzing social interaction predicates. In: Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on. IEEE; 2015. p. 656–662.
11. Ben-Youssef A, Clavel C, Essid S, Bilac M, Chamoux M, Lim A. UE-HRI: a new dataset for the study of user engagement in spontaneous human-robot interactions. In: Proceedings of the 19th ACM International Conference on Multimodal Interaction. ACM; 2017. p. 464–472.
12. Baxter P, Wood R, Belpaeme T. A touchscreen-based 'Sandtray'to facilitate, mediate and contextualise human-robot social interaction. In: Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on. IEEE; 2012. p. 105–106.
13. Olswang L, Svensson L, Coggins T, Beilinson J, Donaldson A. Reliability issues and solutions for coding social communication performance in classroom settings. Journal of Speech, Language & Hearing Research. 2006; 49(5):1058 – 1071. https://doi.org/10.1044/1092-4388(2006/075)
14. Eyben F, Weninger F, Gross F, Schuller B. Recent developments in openSMILE, the munich open-source multimedia feature extractor. In: Proceedings of the 21st ACM international conference on Multimedia. May; 2013. p. 835–838. Available from: http://dl.acm.org/citation.cfm?doid=2502081.2502224.
15. Schuller B, Steidl S, Batliner A. The INTERSPEECH 2009 Emotion Challenge. In: Tenth Annual Conference of the International Speech Communication Association; 2009.
16. Schuller B, Batliner A, Seppi D, Steidl S, Vogt T, Wagner J, et al. The relevance of feature type for the automatic classification of emotional user states: Low level descriptors and functionals. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 2007;2(101):881–884.
17. Kennedy J, Lemaignan S, Montassier C, Lavalade P, Irfan B, Papadopoulos F, et al. Child speech recognition in human-robot interaction: evaluations and recommendations. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. ACM; 2017. p. 82–90.
18. Cao Z, Simon T, Wei SE, Sheikh Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In: CVPR; 2017.
19. Baltrušaitis T, Mahmoud M, Robinson P. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In: Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on. vol. 6. IEEE; 2015. p. 1–6.
20. Lemaignan S, Garcia F, Jacq A, Dillenbourg P. From Real-time Attention Assessment to "With-meness" in Human-Robot Interaction. In: Proceedings of the 2016 ACM/IEEE Human-Robot Interaction Conference; 2016.

UNDERWORLDS: Cascading Situation Assessment for Robots

Séverin Lemaignan[1], Yoan Sallami[2], Christopher Wallbridge[3],
Aurélie Clodic[2], Tony Belpaeme[3], and Rachid Alami[2]

***Abstract*— We introduce UNDERWORLDS, a novel lightweight framework for *cascading spatio-temporal situation assessment* in robotics. UNDERWORLDS allows programmers to represent the robot's environment as real-time distributed data structures, containing both scene graphs (for representation of 3D geometries) and timelines (for representation of temporal events). UNDERWORLDS supports *cascading* representations: the environment is viewed as a set of *worlds* that can each have different spatial and temporal granularities, and may inherit from each other. UNDERWORLDS also provides a set of high-level client libraries and tools to introspect and manipulate the environment models.**

This article presents the design and architecture of this open-source tool, and explores some applications, along with examples of use.

I. INTRODUCTION

UNDERWORLDS is a distributed and lightweight open-source framework[1] that enables robot programmers to build and refine spatial and temporal models of the environment surrounding a robot in real-time. UNDERWORLDS makes it possible to share these world models amongst the software components running on the robot. Additionally, UNDERWORLDS enables users to represent and manipulate *multiple alternatives* to the current, perceived world model in a distributed manner. For instance, the world with some objects filtered out; the world 'viewed' from the perspective of another agent; a hypothetical world resulting from the simulated application of a plan, etc.

A. Distributed Situation Assessment

Anchoring perceptions in a symbolic model suitable for decision-making requires perception abilities and their symbolic interpretation. We call *physical situation assessment* the cognitive skill that a robot exhibits when it represents and assesses the nature and content of its surroundings and monitors its evolution.

Numerous approaches exist, like amodal (in the sense of modality-independent) *proxies* [1], grounded amodal representations [2], semantic maps [3], [4], [5] or affordance-based planning and object classification [6], [7].

UNDERWORLDS is specifically inspired by geometric and temporal reasoners like SPARK (*SPAtial Reasoning & Knowledge*) [8] or TOASTER (*Tracking Of Agents and Spatio-TEmporal Reasoning*) [9]. SPARK acts as a situation assessment reasoner that generates symbolic knowledge from the geometry of the environment with respect to relations between objects, robots and humans. It also takes into account the different perspective that each agent has on the environment. SPARK embeds a modality-independent geometric model of the environment that serves both as basis for the fusion of the perception modalities and as bridge with the symbolic layer [10]. This geometric model is built from 3D CAD models of the objects, furniture and robots, and full body, rigged models of humans. It is updated at run-time by the robot's sensors. Likewise, UNDERWORLDS embeds a grounded amodal model of the environment, updated online from the robot's sensors (sensor fusion).

However, SPARK is a monolithic module that does not support sharing its internal 3D model with other external components. In contrast, UNDERWORLDS focuses on offering a shared and distributed representation of the environment within the robot's software architecture. This also distinguishes UNDERWORLDS from complex cognitive toolkits like KnowRob (as found in OpenEASE [11]). While these tools maintain a spatio-temporal model of the world, this model is internal and not meant to be made widely accessible to other external processes. UNDERWORLDS focuses instead on re-usability and sharing of distributed spatio-temporal models. As such, UNDERWORLDS can be seen as a middleware for spatio-temporal world models and, contrary to KnowRob, it does *not* provide any intrinsic high-level processing or reasoning capability. Such reasoning skills are implemented in loosely-coupled *clients* (see Section III hereafter).

Work on distributed scene graphs [12] has been previously applied to robotics to provide a shared 3D representation of the robot's environment (for instance, the *Robot Scene Graph* [13] or the *Deep State Representation* proposed in [14]). UNDERWORLDS offers a similar distribution mechanism for 3D scene graphs and extends it to temporal representations. Besides, UNDERWORLDS further extends this line of work by providing the ability to create, manipulate and share *multiple alternative worlds*. As an example, these could correspond to filtered or hypothetical *views* on the initial, perceived model of the environment.

B. Representing Alternative States of the World

The components which make use of spatial and temporal models of the environment are usually found in the intermediate layers of robotic architectures, between the low-level perceptual layers, and the high-level decisional layers. They include modules like geometric reasoners (that compute

[1]Author is with Bristol Robotics Lab, University of the West of England, Bristol, United Kingdom severin.lemaignan@brl.ac.uk, [2]Authors are with LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France firstname.surname@laas.fr, [3]Authors are with CRNS, Plymouth University, Plymouth, United Kingdom firstname.surname@plymouth.ac.uk

[1]https://github.com/underworlds-robot/underworlds

spatial and topological relations between objects), motion planners or action recognition modules.

These components exhibit different needs in terms of representation, like different nominal spatial and/or temporal resolutions. For instance, a 3D motion planner would typically use coarse 3D models of surrounding objects to lower the computational load while planning, while a module assessing the visibility of objects might need high-resolution models for accurate 3D visibility testing. This requirement of multiple task-specific representations has been framed as the need for *deep representations* by Beetz [15].

Traditional robotic middlewares, like ROS, are not particularly well suited to deal with these different needs: full geometric data can be represented, but is not first-class citizen: a basic task like displaying a 3D mesh at an arbitrary position is not particularly easy to perform with ROS, requiring the combination of static Collada meshes, a URDF kinematic description, TF broadcasters, and a 3D visualisation tool like RViz. Critically, simultaneously representing and reasoning on alternative states of the environment is not directly feasible.

Representing alternative states is however often highly desirable. For instance, software components manipulating environment models typically perform better if the models are physically consistent. However, low-level perception inaccuracies often introduce hard-to-avoid physical inconsistencies (like detected objects floating in the air, or wrongly inset into other objects). Therefore, a post-process stage (for instance, using a physics simulation engine) is needed to move the objects seen by the robot into physically-correct positions. Implemented with a classical approach (for instance, using ROS TF frames), we would represent an object `book` with two frames: the original frame (e.g.,`book_frame_raw`) and a second one computed by the physics engine (e.g.,`book_frame_corrected`). Such an approach leads to the robot's 3D model being cluttered with multiple frames and does not scale well.

Another example pertains to geometric task planning: a geometric task planner typically needs to reason over hypothetical future states of the environment ("What happens if I move this glass onto that pile of books?"). The planner generates many possible future states, which in turn might require further processing (for instance, running a physics simulation). Such a tool would benefit a flexible representation system, where models are derived from each other, with partial modifications and different timescales.

A third example relates to human-robot interaction scenarios where *perspective taking* is important (a prototypical example being the game 'I spy with my little eye', as implemented in [16]). Perspective taking is a cognitive skill that relies on the ability for an agent to take someone else's point of view to estimate what they see from their perspectives. Perspective taking has previously been implemented in robotics by temporarily placing virtual cameras at eye locations for each of the humans tracked by the robot [17]. While acceptable for simple cases, such an approach does not maintain truly independent spatio-temporal models of the environment for each agent, and in particular, it does not permit the representation of proper false-belief situations. On the contrary, separate, independent world models as implemented by UNDERWORLDS effectively support such a skill, which is an important precursor to research and implement human's mind modelling (i.e., a theory of mind) [18].

Lastly, geometric *pre-supposition accommodation* makes another interesting case for alternative worlds representation. *Pre-supposition accommodation* originally comes from linguistics, where it describes the mechanism by which *context is adjusted [...] to accept [...] a sentence that imposes certain requirements on the context in which it is processed* [19]. In the context of spatio-temporal representations, we call pre-supposition accommodation the ability of an agent to adjust its model so that it matches some contextual constraint. For instance, if A tells B to "catch the red balloon behind you", B might create a representation of an imaginary red balloon, placed behind her, even without actually observing the balloon: B *accommodates* the *pre-supposition* of a red balloon being present behind herself. Endowing robots with this capability has been touched upon by Mavridis et al. within their multi-modal *Grounded Situation Model* [2]. However, to the best of our knowledge, a general framework which would enable robots to accommodate spatial and temporal pre-suppositions by deriving imaginary worlds from existing ones has not been proposed so far.

UNDERWORLDS addresses this need and the main contribution of this work is a generic approach to **represent and share multiple parallel representations of the world**. UNDERWORLDS does so by allowing clients to clone existing worlds, modify them, and re-share them, without the cost of duplicating geometric data (as explained in section II). By organising clients in a network (Figure 1), worlds can be made dependent on each other, resulting in a loosely-coupled modular approach to spatio-temporal world representation that we call *cascading situation assessment*.

II. Design and Architecture

A. Software architecture

Figure 1 depicts a typical UNDERWORLDS topology: a graph (that happens to be an *acyclic* graph on Figure 1, but does not have to be in the general case) of worlds, with clients connecting the worlds to each others.

1) Clients: Software components implementing accessing UNDERWORLDS worlds are called clients. Clients can both read and write onto the worlds they are connected to, and automatically see updates broadcast by other clients connected to the same world. To ensure data consistency, worlds can have many simultaneous readers, but only one writer at a given time.

UNDERWORLDS provides several standard clients (like a 3D visualisation tool or a physics engine simulator). Clients are however typically written by the end users, depending on the needs of one's specific architecture.

2) Worlds: Worlds are effectively distributed data structures composed of a scene graph representing the 3D ge-

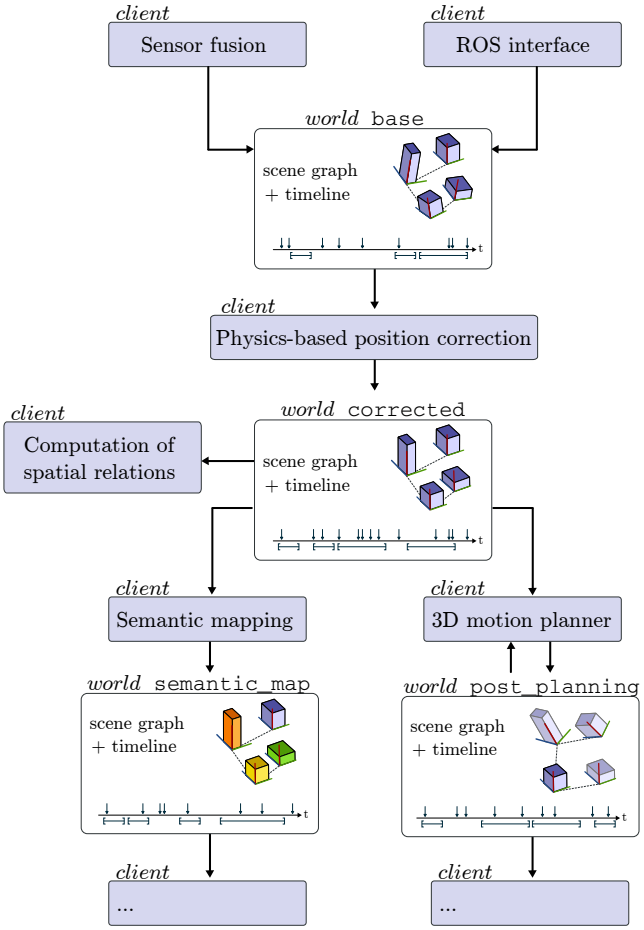

Fig. 1. Schema of a possible UNDERWORLDS network: eight *clients* (user-written & architecture specific; in blue) are sharing environment models through four independent *worlds* (made from joint spatial and temporal models). This architecture enables successive and modular refinement of the models (*cascading* situation assessment), effectively adapted to each client's needs.

ometry of the environment, and a timeline storing temporal events.

While each world is technically independent from all the others, dependencies (and therefore, coupling) arise between worlds from the clients' connections. For instance, filters effectively create a dependency between worlds. On Figure 1, the *Physics-based position correction* client creates a dependency between the world `base` (which represents here the result of raw sensor fusion) and the world `corrected` which would be a physically-consistent copy of `base`. As a result, an UNDERWORLDS network can also be seen as a dependency graph between worlds (where cyclic dependencies are permissible).

This architecture enables what we call *cascading situation assessment*: independent software components (the clients) build, refine and share successive models of the environment by a combination of filtering/transformations steps and model branching. A change performed by one client (for instance, a face tracker updates the pose of the human head) may thereby *cascade* to each of the downstream, dependent worlds.

3) Scenes: Worlds contain both a geometric model and a temporal model. The geometric model is represented as a scene graph. The scene graph has a unique root node, to which a tree of other nodes is parented.

Nodes in an UNDERWORLDS scene graph have three possible types: **objects** that represent concrete physical objects (typically with one or several associated 3D meshes); **entities** that represent abstract entities like reference frames or groups of objects; **perspectives** that represent viewpoints of the scene (like cameras or human gaze).

Every node has a unique ID, a parent, a 3D transformation relative to the parent and an optional name. *Object* nodes optionally store as well pointers to their associated meshes. Importantly, mesh data (or other geometric datasets like point clouds) are *not* stored within the nodes themselves. UNDERWORLDS represents geometric data as immutable data, identified by their hash value (preventing *de facto* data duplication). Nodes only store the hash corresponding to the desired geometric data, and the actual data is pulled from the server by the clients whenever they actually need it (for rendering for instance).

4) Timelines: Complementing the spatial representation encapsulated in the scene graph, each world also stores the world's *timeline*. This data structure is shared and synchronised amongst the clients in the same way as the scene graph. Clients can record and query both *events* (duration-less states) and *situations* in the timeline, i.e., states with a start time and a (possibly open-ended) end time.

B. Distributed spatio-temporal models

UNDERWORLDS is not a monolithic piece of software. Instead, it stands for both a *network of interconnected clients* which manipulate spatial and temporal models of the robot environment (for instance, a motion planner, a object detection module, a human skeleton tracker, etc.), and for a client library that makes it possible to interface existing software components with the network.

Critically, the network is essentially hidden to the client: from the user perspective, the environment model is manipulated as a local data structure (see Listing 1). Modifications to the model are asynchronously synchronised with a central server (the `underworlded` daemon) and broadcast to every other client connected to the same world.

As previously mentioned, worlds are composite data structures comprised of a scene graph and a timeline. These data structures are synchronised using Google's gRPC message passing framework[2], ensuring high throughput, reliability and cross-platform/cross-language support. The UNDERWORLDS API is specifically discussed hereafter, in section III-A.

UNDERWORLDS is meant to broadcast complex environment representations (typically including large geometric datasets, like meshes) in real-time. UNDERWORLDS itself does not perform many CPU intensive tasks (CPU intensive

[2] `http://www.grpc.io/`

processing tasks – sensor fusion, physics simulation, etc.– are performed by the clients themselves) and as such, the performance bottleneck is essentially the network's data throughput. In that regard, one of the simple yet critical optimisations performed by UNDERWORLDS is automatic caching of mesh data. Mesh data are not transmitted when nodes are updated; only a hash value of the mesh data. The client can then request the full data whenever it is actually needed.

C. Time and space complexity analysis

UNDERWORLDS is fundamentally about distributing two datastructures: a scene graph (with nodes representing spatial entities) and a timeline (where events are stored as a flat list). Typical time and space complexities arise from these datastructures. In typical usage scenarios (where the number of nodes or events remain under a few hundred relatively small), the computational load to manipulate these datastructures is however dominated by the actual processings performed by the clients with the data. In the current implementation, scene graphs and timelines are stored in-memory. Were they required, serialization and persistent storage are not anticipated to be difficult to implement.

More interesting is the time complexity of distributing changes across an UNDERWORLDS network. With n the number of worlds and m the number of clients in an UNDERWORLDS network, the worst-case (when every world is a parameter of every client) time complexity of creating or updating a node and propagating the change across the network is $O(n \times m)$ (this effectively corresponds to the UNDERWORLDS server performing $n \times m$ requests to notify clients of the update). The space complexity is the same (as clients own a full copy of the worlds they monitor), except for mesh data whose space and time complexities are $O(1)$ (only the server stores the mesh data).

In the common case of one client performing a full update of a single world (with p nodes) at each time step, the complexity of propagating these changes across the network would be $O(p \times m)$. Figure 2 shows measured propagation time for one change across up to 20 cascading worlds.

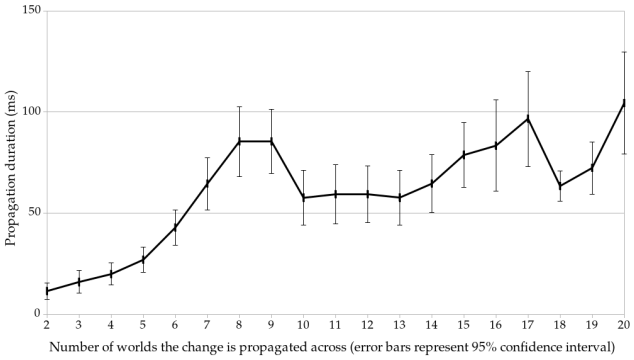

Fig. 2. Propagation times of one change (node creation) across n worlds. The test is performed by running $n-1$ pass-through filters that monitor one world and replicate any changes into the next world. Durations measured over 20 runs, performed on a 8 core machine.

III. API & CLIENTS

A. API

As mentioned, UNDERWORLDS uses Google's gRPC as message passing protocol. The protocol is explicitly defined (using the *protocol buffers*[3] interface definition language), and bindings to various languages and platforms can be automatically generated from the protocol definition file (as of Jan 2018, gRPC can generate bindings for C, C++, C#, Node.js, PHP, Ruby, Python, Go and Java, on Windows, Mac, Linux and Android). The cross-platform/cross-language support of gRPC is especially welcome in the academic context, as it offers ease and flexibility to plug a variety of pre-existing components into an UNDERWORLDS network.

[3] `https://developers.google.com/protocol-buffers/`

However, the gRPC message passing layer is low-level with respect to the typical use of UNDERWORLDS (manipulation of asynchronous, distributed spatio-temporal models of the robot environment). In particular, the asynchronous fetching (and conversely, remote updating) of nodes and time-related objects is typically hidden from the user, and managed instead by the UNDERWORLDS client library.

UNDERWORLDS currently offers such a high-level client library for Python only (a C++ library is under development). Listing 1 gives a complete example of an UNDERWORLDS client performing simple filtering: the client continuously listens for changes in an input world, removes some objects (in this case, items whose volume is below a threshold), and forwards all other changes to an output world, effectively making the output world a copy of the input world with all smaller objects removed.

```
1  import underworlds
2
3  # by default, connect to the server on localhost
4  with underworlds.Context("small_object_filter") as ctx:
5
6      in_world = ctx.worlds["world1"]
7      out_world = ctx.worlds["world2"]
8
9      while True:
10
11         in_world.scene.waitforchanges()
12
13         for node in in_world.scene.nodes:
14             if node.volume > THRESHOLD:
15                 out_world.scene.nodes.update(node)
```

Listing 1: Example of a simple yet complete UNDERWORLDS filter, written in Python: the client connects to the UNDERWORLDS network, blocks until the world `world1` changes, and only propagate nodes that match the condition to the world `world2`.

B. Standard Clients

The UNDERWORLDS package provides several standard clients to perform common tasks on UNDERWORLDS networks.

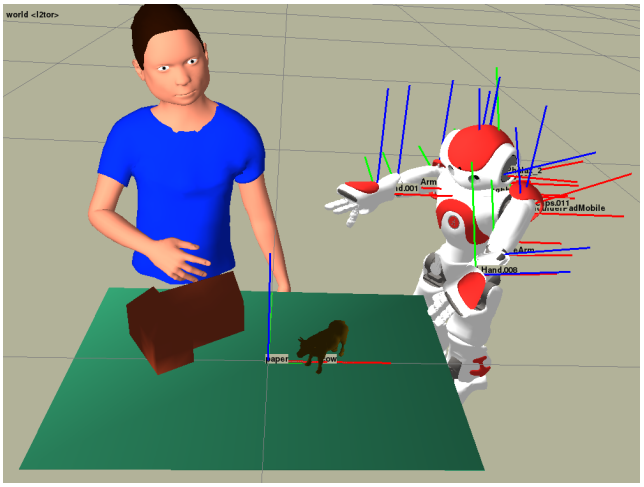

Fig. 3. Screenshot of the `uwds view` 3D visualisation and manipulation client. In this particular example, the 3D meshes have been pre-loaded using `uwds load`. Their positions are then updated at run-time using the robot's sensors and proprioception (joint state).

1) 3D Visualisation and manipulation: Interestingly, while UNDERWORLDS deals with 3D geometries and scenes, it does represent 3D entities purely as data structures; no visual representation is involved (and as such, the UNDERWORLDS server and core libraries do not depend on any graphics library like OpenGL). However, for all practical purposes, the ability to visualise the content of a scene is desirable. UNDERWORLDS provides a standard client, `uwds view`, that performs real-time 3D rendering of worlds, using OpenGL (Figure 3).

This tool also supports basic object manipulations (translations, rotations), that are broadcast to the other UNDERWORLDS clients connected to the same world.

Assets loading: Often, objects manipulated by the robot have known meshes with corresponding CAD models that can be conveniently pre-loaded. In these cases, UNDERWORLDS provides a tool, `uwds load`, that loads a mesh into a UNDERWORLDS network (and optionally, creates a node) from a large range of 3D formats (including Collada, FBX, OBJ, Blender)[4].

2) Physics simulation: When perception modules provide objects localisation, the physical consistency of the locations is not typically enforced. For instance, objects that are supposed to lay on a table might be slightly above (or inset into) the table; or when dropping an item into a box, the robot can not update the location of the item anymore as it becomes occluded.

These issues can be alleviated by relying on a physics simulation to stabilise the position of objects: natural physics (including gravity) are simulated for a short amount of time (up to one second) ahead of time, and the objects' positions are updated accordingly. To this end, UNDERWORLDS provide a standard filter, the `physics_filter`, based on the Bullet RT physics simulation and the `pybullet`[5] library. It generates an output world that mirrors its input world after a specific duration of physics simulation, the physical properties of objects (including mass, friction, inertia) being provided from standard URDF descriptions.

[4]The underlying import capability is provided by the ASSIMP library. `http://assimp.sourceforge.net/`

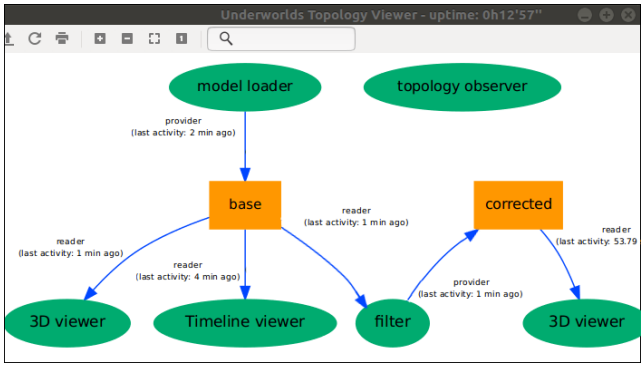

Fig. 4. Screenshot of the network topology introspection tool, with arbitrary examples of worlds (represented as boxes) and clients (ellipses). CLients are connected to the worlds either as *readers* or *providers* of data. UNDERWORLDS introspection features make it possible to also visualise when clients were last active.

3) Introspection and debugging: UNDERWORLDS provides a range of tools to inspect a running network. Graphical tools (`uwds explorer` and `uwds timeline`, see Figure 4) provide a user-friendly overview of the system's graph with the connections between the clients and the worlds, as well as their activity.

Specialised command-line tools are also available to list the worlds and their content (`uwds ls`) at run-time, or to display detailed information for a specific node (`uwds show`).

4) Interface with ROS: UNDERWORLDS is meant to integrate as easily as possible into existing robot architectures, and interfaces transparently with ROS' TF frame system through the `uwds tf` client.

The `uwds tf` client continuously monitors the ROS TF tree, and mirrors TF frames as nodes in the desired UNDERWORLDS world. A node is first created if none matches a given TF frame, and its transformation is subsequently updated, mirroring the TF frame. A regular expression can be provided to only mirror a subset of the TF tree into UNDERWORLDS.

Currently, the process is unidirectional: the `uwds tf` client performs TF to UNDERWORLDS updates, but not the reverse.

C. Spatial Reasoning and Perspective Taking

Spatial reasoning [20] is a field in its own right, and has been used for natural language processing for applications such as direction recognition [21], [22] or language grounding [23]. Other examples in human-robot interaction include

[5]`https://pybullet.org/`

Ros et al. [17], [16] which has recently been integrated into a full architecture for autonomous human-robot interaction [10].

UNDERWORLDS provides an exemplary client (`spatial_relations`) to compute both allo-centric (independent of the viewpoint like `isIn` or `isOn`) and ego-centric (i.e., viewer-dependent, like `inFrontOf` or `leftOf`) spatial relations between objects. Other libraries, like QSRLib [24], that implement computational models of Qualitative Spatial Relations, could be trivially combined with UNDERWORLDS to provide more advanced geometric analysis. Future developments will also include the results of the more basic research on spatio-temporal reasoning for robotics, led by de Leng and Heintz [25].

UNDERWORLDS also implements an efficient algorithm to assess object visibility from a specific viewpoint (i.e., from a given *perspective* node). The algorithm (color picking) enables fast (single pass) computation of the visibility of every object in the scene, while providing control regarding how many pixels should be actually visible for the object to be considered globally visible. The command-line tool `uwds visibility` returns the list of visible objects from the point of view of each camera in a given world, and UNDERWORLDS also provides the helper class `VisibilityMonitor` to programmatically access visibility information.

When integrated into a filter node, visibility computation allows easy creation of new worlds representing the estimated perspectives of the different agents.

IV. Application Example: Perspective-aware joint actions

UNDERWORLDS is being used within the large European project MuMMER[6] for service robots to compute visibility and knowledge about objects, places and agents within a mall environment.

We present here a simplified scenario, yet representative of situations which are processed in real-time by MuMMER robots: two humans and a robot are looking at a table and have to coordinate joint actions (pick and place). One object on the table (the green box in Figure 5) is only visible to one human and the robot, but hidden to the second human. The robot needs to take into account this fact to generate appropriate and legible joint manipulation actions. Figure 5 illustrates the topology of the UNDERWORLDS network that we use to this end.

A first client, *static_env_provider*, provides the environment models and allows to build a first `ENV` world where static objects, furnitures and walls are present. Then, three worlds cascade through three (independent) clients: *robots_state_monitor* augments `ENV` with the robot state (using underneath the ROS robot state publisher node) and broadcast a new world `ENV_ROBOTS`. *objects_monitor* then recognises and adds the dynamic objects (using `ar_track_alvar`[7]). *humans_monitor* finally detects and

[6] `http://www.mummer-project.eu`

[7] `http://wiki.ros.org/ar_track_alvar`

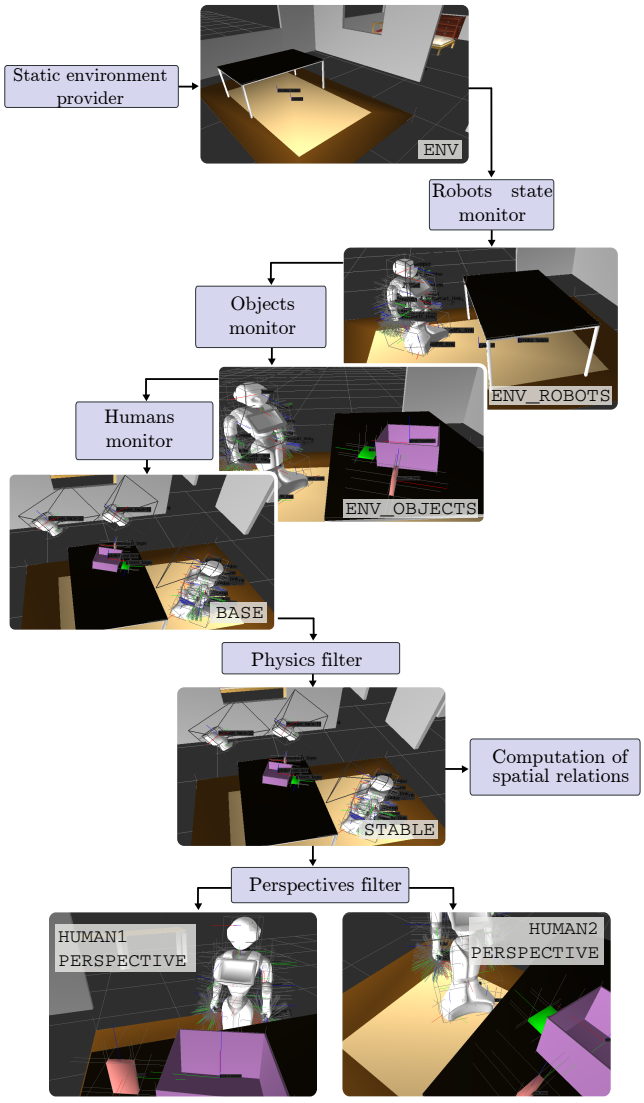

Fig. 5. Schema of the UNDERWORLDS architecture used in the MuMMER project. Clients read and generate the worlds `ENV` $\rightarrow$ `ENV_ROBOT` $\rightarrow$ ... $\rightarrow$ `HUMAN*_PERSPECTIVE`. The last two worlds `HUMAN{1,2}_PERSPECTIVE` represent the immediate visual perspective of each of the humans, as well as their past visual perceptions. As such, they are the *visual memories* of the humans, that the robot can rely on when making decisions.

continuously updates the humans poses (using [26]). It broadcasts a world called `BASE` that contains as a result the static environment, the robots, the dynamic objects and the detected humans.

The world `BASE` goes through a *physics filter* client (as explained in section III-B.2) to obtain the `STABLE` world where all elements are present with physically-consistent locations. This physically-correct world is used by the *computation_of_spatial_relations* client to compute spatial relations such as `onTop`, `isIn` or `isAbove` (see Section III-C).

The world `STABLE` is also used by a *perspectives_filter* client to compute the different visual perspectives of each agent (in our case: human 1, human 2 and the robot itself).

In addition to a 3D rendering of the input world from the perspective of the agent, it aggregates the history of what was visible to the agent at a given point in time. As such, it does not only offer a snapshot of the agent visual perspective at the current time but also acts as the visual memory of each agent.

With this network, the robot can easily compute that an object on the table is only seen by the human 1 and not the human 2; additionally, if human 1 moves in a position where the object is not visible anymore to him, the *perspective_filter* will maintain the knowledge that the human had seen it (and keep the last position where it has been seen).

UNDERWORLDS makes it possible to implement such a geometric reasoning pipeline in a fully decoupled way, and each intermediary world can be easily introspected at run-time. This example shows how UNDERWORLDS facilitates the implementation and debugging of complex spatio-temporal reasoning pipelines.

We are currently deploying a similar network in the framework of the European project MuMMER where a Pepper robot handles interactive situations in a large shopping centre in Finland. One of the situation is a guiding task where Pepper help people to find their route by pointing them landmarks and explaining them how to reach a destination. To be effective, this helping behaviour needs to be aware of the visual perspective of the human. UNDERWORLDS facilitates the implementation of such a spatio-temporal reasoning pipeline, where perception and high-level reasoning (including complex, human-aware reasoning) have to be tightly integrated. Because of the decoupling of each of the clients in the network, UNDERWORLDS also practically supports software development spread across multiple partners in different countries, with different expertise.

V. Discussion and Conclusion

A. Relation to existing robotic middleware

Like traditional robotic middleware, UNDERWORLDS offers a form of distributed computation based on message passing. However, it distinguishes itself from existing middlewares (including ROS extensions like DyKnow [27]) in significant ways. Most importantly, UNDERWORLDS purposefully does not offer any *general* capability to distribute computation and data streams amongst independent components: it focusses specifically on distributing environment models, both spatial (geometric models) and temporal (events and situations). In that sense, UNDERWORLDS really is a *distributed datastructure* that addresses the specific needs of spatio-temporal modelling, including the modelling of hypothetical, alternative world models, something that traditional middlewares like ROS do not address adequately. Second, and as presented above, UNDERWORLDS offers specific mechanisms for the representation and manipulation of alternative world models that are not directly achievable with traditional tools.

While using standard middleware as *underlying transport* for UNDERWORLDS would be technically feasible and relatively easy to implement, it does not offer any clear advantage over lighter and dedicated message passing libraries like ZeroMQ or gRPC (the later being the one used by UNDERWORLDS).

B. Future work

As illustrated in section IV, UNDERWORLDS is already deployed and used on the field. Several features are however still under development.

1) Representation capabilities: as presented in section II, the current version of UNDERWORLDS allows to represent *objects*, *abstract entities* like groups and *perspectives*. *Fields* are also part of the UNDERWORLDS design, but are not yet implemented. Fields are commonly used to represent continuously-valued spatial entities. Fields might or might not be spatially bounded. Examples include the working space of a robot arm (spatially bounded), the field of view of a camera (spatially bounded), proxemics (potentially unbounded). We plan to represent fields in UNDERWORLDS using the memory-efficient octomaps [28] or NDT-OM maps [29]. Similarly to geometric data,these datastructures will not be directly stored with the nodes (nodes will refer to them through handles), but unlike geometric data, they will not be treated as immutable datasets by the server, permitting real-time updates.

Representation of uncertainty: currently node positions are stored as 4×4 transformation matrices, relative to the node parent. This representation is efficient, and conveniently matches traditional representation systems (including ROS TF frames or OpenGL transformations). However, the explicit management of uncertainties is instrumental to many robotic applications, and we plan to add full support for position uncertainties to UNDERWORLDS. We plan to add this support by adding a pose covariance matrix to the nodes, and equipping the different UNDERWORLDS helper tools with corresponding support (like covariance ellipses visualisation in `uwds view`).

2) Implementation and Integration: we plan to continue to improve the integration of UNDERWORLDS into existing software architectures. A short-term goal is to provide excellent C++ support, with a high-level, user-friendly C++ client library. This is critical for a broader adoption of UNDERWORLDS within the robot community. Support for other languages might follow, depending on demands and open-source contributions.

C. Conclusion

We have introduced UNDERWORLDS, a novel framework for shared and composable spatio-temporal representations of a robot's world. The key contributions of our approach are: a composite data structure for environment representation within a robotic software architecture, made of a scene graph and a timeline; a mechanism to efficiently and transparently share this data structure amongst a set of clients (the software modules of the robot); a cascading architecture permitting the explicit of representation of alternative states of the world while maintaining a network of dependencies.

We have additionally presented a concrete instantiation of a system relying on UNDERWORLDS for its representation needs, and we have sketched future directions of development.

We believe this work can practically support existing robotic architectures with state-of-the-art spatio-temporal representation capabilities. We also hope that this line of research can lead to a better understanding of the representation needs of modern robotic systems, and participate to the emergence of a possible common representation platform for robots, building on previous formalisation efforts like the RSG-DSL domain specific language [30].

ACKNOWLEDGMENT

This work has been supported by the EU H2020 Marie Skłodowska-Curie Actions project DoRoThy (grant 657227), the EU H2020 MuMMER project (grant 688147) and the EU H2020 L2TOR project (grant 688014).

REFERENCES

[1] H. Jacobsson, N. Hawes, G.-J. Kruijff, and J. Wyatt, "Crossmodal content binding in information-processing architectures," in *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*. New York, NY, USA: ACM, 2008, pp. 81–88.

[2] N. Mavridis and D. Roy, "Grounded situation models for robots: Where words and percepts meet," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[3] A. Nchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915 – 926, 2008.

[4] C. Galindo, J. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 955–966, 2008.

[5] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Rühr, M. Tenorth, and M. Beetz, "Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, September, 25–30 2011.

[6] C. Lörken and J. Hertzberg, "Grounding planning operators by affordances," in *International Conference on Cognitive Systems (CogSys)*, 2008, pp. 79–84.

[7] K. Varadarajan and M. Vincze, "Ontological knowledge management framework for grasping and manipulation," in *IROS Workshop: Knowledge Representation for Autonomous Robots*, 2011.

[8] E. A. Sisbot, R. Ros, and R. Alami, "Situation assessment for human-robot interactive object manipulation," in *2011 RO-MAN*, July 2011, pp. 15–20.

[9] 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 *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, 2014, pp. 1103–1109.

[10] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, and R. Alami, "Artificial cognition for social human-robot interaction: An implementation," *Artificial Intelligence*, 2016.

[11] M. Beetz, M. Tenorth, and J. Winkler, "Open-EASE – a knowledge processing service for robots and robotics/ai researchers," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1983–1990.

[12] M. Naef, E. Lamboray, O. Staadt, and M. Gross, "The blue-c distributed scene graph," in *Proceedings of the workshop on Virtual environments 2003*. ACM, 2003, pp. 125–133.

[13] S. Blumenthal, H. Bruyninckx, W. Nowak, and E. Prassler, "A scene graph based shared 3d world model for robotic applications," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 453–460.

[14] P. Bustos, L. J. Manso, J. P. Bandera, A. Romero-Garcés, L. V. Calderita, R. Marfil, and A. Bandera, "A unified internal representation of the outer world for social robotics," in *Robot 2015: Second Iberian Robotics Conference*. Springer, 2016, pp. 733–744.

[15] M. Beetz, D. Jain, L. Mösenlechner, and M. Tenorth, "Towards performing everyday manipulation activities," *Robotics and Autonomous Systems*, vol. 58, no. 9, pp. 1085–1095, 2010.

[16] R. Ros, S. Lemaignan, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken, "Which one? grounding the referent based on efficient human-robot interaction," in *19th IEEE International Symposium in Robot and Human Interactive Communication*, 2010.

[17] R. Ros, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken, "Solving ambiguities with perspective taking," in *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2010, pp. 181–182.

[18] S. Lemaignan and P. Dillenbourg, "Mutual modelling in robotics: Inspirations for the next steps," in *Proceedings of the 2015 ACM/IEEE Human-Robot Interaction Conference*, 2015.

[19] K. Von Fintel, "What is presupposition accommodation, again?" *Philosophical perspectives*, vol. 22, no. 1, pp. 137–170, 2008.

[20] J. O'Keefe, *The Spatial Prepositions*. MIT Press, 1999.

[21] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *HRI*, 2010, pp. 259–266.

[22] C. Matuszek, D. Fox, and K. Koscher, "Following directions using statistical machine translation," in *Proceedings of the International Conference on Human-Robot Interaction*. ACM Press, 2010.

[23] S. Tellex, "Natural language and spatial reasoning," Ph.D. dissertation, MIT, 2010.

[24] Y. Gatsoulis, M. Alomari, C. Burbridge, C. Dondrup, P. Duckworth, P. Lightbody, M. Hanheide, N. Hawes, D. Hogg, A. Cohn *et al.*, "Qsrlib: a software library for online acquisition of qualitative spatial relations from video," Tech. Rep., 2016.

[25] D. de Leng and F. Heintz, "Qualitative spatio-temporal stream reasoning with unobservable intertemporal spatial relations using landmarks." in *AAAI*, 2016, pp. 957–963.

[26] V. Khalidov and J.-M. Odobez, "Real-time multiple head tracking using texture and colour cues," Idiap, Idiap-RR Idiap-RR-02-2017, 2 2017.

[27] D. de Leng and F. Heintz, "DyKnow: A Dynamically Reconfigurable Stream Reasoning Framework as an Extension to the Robot Operating System," in *IEEE Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2016, pp. 55–60. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-132266

[28] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[29] J. P. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, "3d normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1627–1644, 2013.

[30] S. Blumenthal and H. Bruyninckx, "Towards a domain specific language for a scene graph based robotic world model," *arXiv preprint arXiv:1408.0200*, 2014.