ROS for Human-Robot Interaction

Youssef Mohamed[1,2] and Séverin Lemaignan[1]

Abstract— **Integrating real-time, complex social signal processing into robotic systems – especially in real-world, multi-party interaction situations – is a challenge faced by many in the Human-Robot Interaction (HRI) community. The difficulty is compounded by the lack of any standard model for human representation that would facilitate the development and interoperability of social perception components and pipelines. We introduce in this paper a set of conventions and standard interfaces for HRI scenarios, designed to be used with the Robot Operating System (ROS). It directly aims at promoting interoperability and re-usability of core functionality between the many HRI-related software tools, from skeleton tracking, to face recognition, to natural language processing. Importantly, these interfaces are designed to be relevant to a broad range of HRI applications, from high-level crowd simulation, to group-level social interaction modelling, to detailed modelling of human kinematics. We demonstrate these interfaces by providing a reference pipeline implementation, packaged to be easily downloaded and evaluated by the community.**

I. Introduction

Social signal processing (both signal detection, and signal interpretation) is a fundamental task in Human-Robot Interaction (HRI). Traditionally, this task is approached through social signal processing *pipelines*: a combination of software modules, that each implement a stage of signal processing, and feed their output to the next module. This pipeline-based approach is common in robotics, for instance for 2D navigation[1], or 3D image processing[2]. The Robot Operating System (ROS) [18] has played an instrumental role in enabling quick and iterative design and implementation of such processing pipelines, by standardizing loosely coupled data streams (ROS *topics*) and corresponding datatypes (ROS *messages*). And indeed, ROS is today used pervasively in the academic and industrial robotic communities, as the go-to solution to create real-time data processing pipelines for complex, real-world sensory information.

Surprisingly, no single effort has been successful in creating similar, broadly accepted interfaces and pipelines for the HRI domain. As a result, many different implementations of common tasks (skeleton tracking, face recognition, speech processing, etc.) cohabit with their own set of interfaces and conventions. More concerning for the development of decisional architectures for interactive autonomous robots, the existing software modules are not designed to work together: a skeleton tracker would typically estimate 3D poses of bones entirely independently of eg a facial expression recognizer. As a consequence, *matching* a 3D body pose to its corresponding face requires a third-party module, whose role is to *track* detected skeletons, detected faces (also in case of temporary occlusions), and associate them. How this association is then published and shared with the rest of the architecture is effectively implementation-dependent. Note that we take here the example of matching bodies to facial expression, but the same could be said of voice processing, speech, gaze estimation, head pose, etc.

The lack of a ROS standard for HRI can be explained both by the relative lack of maturity of some of the underlying detection and processing algorithms (for instance, 3D skeleton tracking is a less mature technology than simultaneous localization and mapping (SLAM) algorithms used in 2D navigation pipelines), but also by the sheer complexity of HRI pipelines. Besides the body/face matching issue mentioned above, we can also mention the highly variable *scale* (or *granularity*) at which humans are required to be modeled, depending on the application: from simple, abstract 3D positions in high-level crowd simulation, to group-level social interaction modelling (that would for instance require accurate gaze modelling), to accurate modelling of human kinematics, for eg kinaesthetic teleoperation or Learning for Demonstration. Also, contrary to most of the objects and situations traditionally encountered in robotics, humans are bodies that are typically not known prior to runtime, and are highly dynamic: it is commonly expected that they will appear and disappear from the robot sensory space multiple times during a typical interaction. This transient nature causes various issues, including a need for robust tracking, re-identification, managing a history of known people, etc.

In order to provide robust, complete foundations on which to address these issues, we present in this article the *ROS4HRI* framework. Our contribution with ROS4HRI is:

1) the specification a new ROS-based *representation model for humans*, both appropriate for a broad range of HRI applications (from a single individual to crowds), and accounting for the design choices made by state-of-art social signal processing tools;
2) the specification of ROS conventions and data types to enable the future development of multi-modal and interoperable social signal processing pipelines.

Alongside these two specifications, we also present an open-source implementation of the ROS4HRI framework, that currently covers a subset of the specifications, namely the 3D tracking and matching of skeletons and faces in groups of up to about 10 people.

[1] Youssef Mohamed and Séverin Lemaignan are with Bristol Robotics Laboratory, University of the West of England, Bristol, UK `severin.lemaignan@brl.ac.uk`

[2] Youssef Mohamed is also with KTH Royal Institute of Technology, Stockholm, Sweden `ymo@kth.se`

[1] http://wiki.ros.org/navigation

[2] http://wiki.ros.org/ecto

The main open-source code repository can be found here: github.com/ros4hri/ros4hri.

The remaining of the article is structured in the following way: we review previous work pertaining to ROS and HRI; we then introduce our human model (made of four components: the *body*, the *face*, the *voice* and the *person*); we present the ROS specification of our model: a combination of a limited set of new HRI ROS messages with ROS topic and frame naming conventions; we then present a reference implementation of a ROS4HRI pipeline, validated on a small dataset of naturalistic social interactions.

II. Related work

We look first into some significant *non-ROS* social signal processing approaches; we then cover the (limited) early attempts at creating ROS interfaces for HRI; finally, we discuss a few ad-hoc projects which used ROS for HRI, without attempting to build a generic, application-agnostic framework out of it.

Note that this paper is not about *specific* algorithms to perform multi-modal social signal processing in robotics, but rather on building integrated pipelines of algorithms. We can refer the interested reader to [4] for an introduction to social signal processing and to the relevant surveys already published on specific social signals processing techniques.

A. Approaches to social signal processing in HRI

Several frameworks have been developed over the years for HRI; [8] introduce the human-robot interaction operating system (HRI/OS). HRI/OS is an architecture that allows cooperation between humans and robots. The HRI/OS supports peer-to-peer dialogue, and their architecture introduces a way to assign tasks to agents. HRI/OS does not explicitly model humans, however, but introduced nonetheless the idea of creating a framework for HRI.

The LAAS architecture for social autonomy [15] is another framework featuring real-time modelling of human interactors. SHARY, their architecture controller, aims at enhancing the collaboration between humans and robots by introducing a layered architecture for decision planning. The framework is focused on one-to-one interaction and models the human's position and gaze, which had a direct effect on the decision planning process. While featuring advanced human modelling capabilities, the LAAS architecture is not designed for interoperability with other systems and does not rely on standard datatype.

In the Social Signal Interpretation (SSI) framework [21], social signals are recorded, analyzed and classified in real-time. The patch-based design of the SSI allows numerous types of sensors to be integrated with the ability for all of them to work in parallel and synchronize the input signals. SSI also supports the use of machine learning models, as it has a graphical user interface that aids in the process of annotating the data and then integrating the models created in the data extraction process. It does not however address the question of broad modularity and interoperability.

Lastly, the social perception capabilities of the NAOqi framework (as found in the Softbank Nao and Pepper robots [17]) have to be mentionned: while not open-source, and as such, not available outside of the Softbank robots, the NAOqi framework features advanced social perception capabilities like people recognition, gaze and expression monitoring, engagement estimation. Several of the design choices of our ROS4HRI framework (like the use of both transient user ID and permanent person ID to keep track of people even when they are temporarily not detected anymore) have been influenced by NAOqi.

B. ROS standards for HRI

Only a few attempts have been made in the literature to utilize ROS as the underlying technology for social signal processing, often focusing on one type of social signals, and ignoring the challenge of modalities fusion.

To the best of our knowledge, only two ROS projects have attempted to create a stand-alone toolset for HRI: the `people`[3] package, originally developed by Pantofaru in 2010-2012 (last code commit in 2015), and the `cob_people_perception`[4] package [3], developed in 2012-2014 in the frame of the EU project ACCOMPANY (and still maintained).

Neither of these two attempts is however generic in the sense that they propose a complete, multi-modal, technology-agnostic approach: the `people` package had a narrow scope (leg tracking and face tracking), and the `cob_people_perception` stack is mainly built around the Kinect hardware and NITE software library. However, some of the HRI ROS messages we introduce hereafter have roots in these two early attempts.

On the matter of representing the human body using ROS conventions, we draw our naming conventions from the work done in humanoid robots. Specifically, the ROS REP-120[5] partially defines a naming convention for humanoid robots that we follow here to a large extend.

The Human-Robot Interaction toolkit [14] (`HRItk`) is another ROS package for speech processing. This is done by integrating several natural language modules, like speech detection and recognition, natural language understanding, and dialogue state analysis. `HRItk` also has two basic models for gesture recognition and gaze tracking, both of which were basic concepts and are not maintained in the toolkit. It does not cover the uses of other social signals, like body language and facial expressions. Nonetheless, the toolkit provided an efficient architecture for NLP using ROS, and the bases of other architectures in the literature [16], [22].

C. Ad-hoc ROS-based pipelines for HRI

While there are few conclusive efforts to standardized HRI tools in ROS, ROS has been used in several complex systems with HRI applications. For example, the STRANDS project has been covering a range of issues in the HRI field, from

[3] https://wiki.ros.org/people

[4] http://wiki.ros.org/cob_people_detection

[5] https://www.ros.org/reps/rep-0120.html

world mapping to human activity recognition. [10] reports for instance on their ROS-based integration of a robot in physical therapy sessions for older adults with dementia. While broadly successful, the authors point out that this experiment leads to better group-level modelling of human-robot interactions.

The POETICON++ project has designed another ROS-based architecture for HRI applications with a focus on natural language processing [1], [20], similar in that sense to the multi-modal framework created by [5], which utilized ROS to communicate between the natural language processing modules and the ontology and state publisher. The scope of the paper did not go beyond giving verbal commands to a robotic arm to perform a wielding task hence did not take any other inputs from the human.

The SHRI framework designed by [12] proposed an architecture which separates the system into two main parts: social perception and social task controller. Although the architecture proposed discussed both elements, the social perception element was only discussed in a general manner, without mentioning plans for standardizing the process.

A similar architecture was discussed by [11], with three separate elements: perceptual, cognitive and behavioural systems. Both the cognitive and behavioural systems are responsible for processing the data collected from the perceptual system and generating actions based on it. Furthermore, the perceptual system has been discussed in some level of detail and an attempt to standardize the processed data has been made by using XML files in ROS. Yet, the paper only discussed gaze and speech in the perceptual system of the robot, and the attempt to standardize the data did not account for multiple people in groups. Furthermore, as most other frameworks created for HRI, this framework focused more on generating human-like behaviours rather than analysing them.

These several examples of custom architectures illustrate the need to standardize human-robot interaction pipelines to accelerate their development and significantly increase code reusability across projects. We present the next two specifications to support the endeavour: the specification of a generic multi-modal *human model* (Section III), and a set of ROS conventions to facilitate interoperability (Section IV).

III. The ROS4HRI human model

A. The four human identifiers

To accommodate existing tools and techniques used to detect and recognize humans, the representation of a person is built on a combination of 4 unique identifiers: a face, a body, a voice and a person identifier. These four identifiers are not mutually exclusive, and depending on the requirements of the application, the available sensing capabilities, and the position/behaviour of the humans, only some might be available for a given person, at a given time (Figure 1).

a) Face identifier: The face identifier is a unique ID (UUID) that identifies a detected face. This ID is typically generated by the face detector/head pose estimator upon face detection. There is a one-to-one relationship between this face ID and the estimated 6D pose of the head represented as a ROS TF frame named `face_<faceID>` (cf below for details regarding the face frame conventions). Importantly, this ID is **not persistent**: once a face is lost (for instance, the person goes out of frame), its ID is not valid nor meaningful anymore. To cater for a broad range of applications (where re-identification might not be always necessary), there is no expectation that the face detector will attempt to recognise the face and re-assign the same face ID if the person reappears.

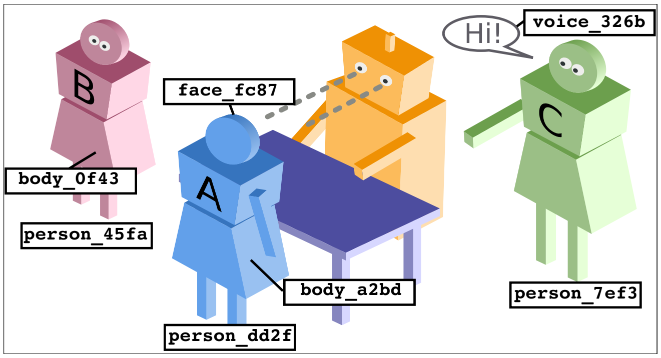

Fig. 1: In this situation: A is facing the robot: A gets a unique `faceID`, a unique `bodyID`, and a unique `personID`; B's body is visible to the robot, but not the face: B only gets a `bodyID` and `personID`; C is not seen, but heard: C gets a `voiceID` and a `personID`.

At any given time, the list of tracked faces is published under the `humans/faces/tracked` topic.

b) Body identifier: The body identifier is similar to the face ID, but for a person's skeleton. It is typically created by the skeleton tracker upon detection of a skeleton. Like the face ID, the body ID is **not persistent** and is valid only as long as the specific skeleton is tracked by the skeleton tracker which initially detected it. The corresponding TF frame is `body_<bodyID>`, and TF frames associated with each of the body parts of the person, are suffixed with the same ID (cf below).

The list of tracked skeletons is published under the `humans/bodies/tracked` topic.

c) Voice identifier: Likewise, a voice separation module should assign a unique, non-persistent, ID for each detected voice. Tracked voices are published under the `humans/voices/tracked` topic.

d) Person identifier: Finally, the person identifier is a unique ID **permanently** associated with a unique person. This agent ID should be assigned by a module able to perform person identification (face recognition module, voice recognition module, sound source localization + name, identification based on physical features like height/age/gender, person identification based on pre-defined features like the colour of the clothes, etc.) This ID is meant to be persistent so that the robot can recognize people across encounters/sessions.

As soon as a face, a body or a voice is detected, the robot can infer the presence of a person, and therefore a person ID

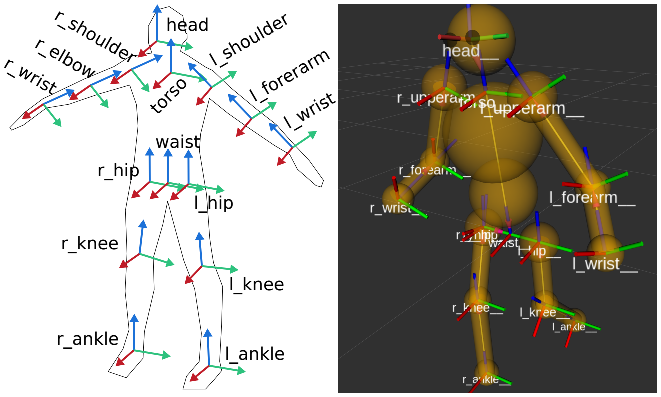

Fig. 2: Left: the 15 links defined on the human body. Frames orientations and naming are based on REP-103 and REP-120. Right: the URDF kinematic model, viewed in RViz.

must be created and associated with that face/body/voice. As person IDs are permanent, that ID will permanently remain in the robot's knowledge.

When meaningful, a TF frame must be associated with the agent ID, named `person_<personID>`. Due to the importance of the head in human-robot interaction, the `person_<personID>` frame is expected to be placed as close as possible to the head of the human. If neither the face nor the skeleton is tracked, the `person_<personID>` frame might be located to the last known position of the human or removed altogether if no meaningful estimate of the human location is available. We detail below the rules associated to the `person_<personID>` frame.

B. Human kinematic model

URDF[6] is the XML-based language used by ROS to represent kinematic models. Besides visualization, URDF models are used by several ROS tools to reason about the kinematic of systems (for instance, for motion planning or grasp planning). In order to leverage these tools, we adopt a URDF-centric approach to human kinematics.

However, unlike robots (whose kinematic models are usually fixed, and known beforehand), humans anatomies do vary from one person to the other, reflecting her/his individual height, weight, etc. We account for these differences by generating on-the-fly custom URDF models every time a person is detected[7], using the person's observed height as the main parameter, from which other dimensions (like the shoulder-to-shoulder width, the waist width, and the length of the limbs) are derived, based on standard models of anthropometry.

The generated URDF model is then published on the ROS parameter server (using the `bodyID` unique identifier), making it available to the rest of the ROS network. The URDF model is used in combination with the computed joint state of each tracked body part to then generate a kinematically-sound, real-time 3D model of the person (Figure 2).

IV. ROS SPECIFICATION

A. Topics structure

Our implementation exposes social signals using a specific structure of ROS topics, and introduces a limited number of new ROS messages.

We introduce the following rules to present human perceptions in a ROS system:

1) all topics are grouped under the global namespace `/humans`
2) five sub-namespaces are available:
   - `/humans/faces`
   - `/humans/bodies`
   - `/humans/voices`
   - `/humans/persons`
   - `/humans/interactions`
3) the first four (`/faces`, `/bodies`, `/voices`, `/persons`) expose one sub-namespace per face, body, voice, person detected, named after the corresponding id: for instance, `/humans/faces/<faceID>/`. In addition, they expose a topic `/tracked` where the list of currently tracked faces/bodies/voices/persons is published.
4) the `/humans/interactions` topic exposes group-level signals, including gazing patterns and social groups.

The structure of each sub-namespace is presented in Table I.

B. The `hri_msgs` ROS messages

Table II lists the newly introduced ROS messages for HRI. They are regrouped in the `hri_msgs`[8] ROS package.

C. Frame conventions

The ROS4HRI specifies several TF frames to spatially represent a human. Where meaningful, the frames follow the conventions set out in the ROS *REP-103 – Standard Units of Measure and Coordinate Conventions*[9] and *REP-120 – Coordinate Frames for Humanoid Robots*[10].

1) Body frames: Figure 2 shows the 15 frames defined on the human skeleton. The `waist_<bodyID>` is collocated with the body's root frame, `body_<bodyID>` (where `<bodyID>` stands for the unique body identifier). The origin of this frame is located at the midpoint between the two hips, and the parent of this frame would typically be the sensor frame used to estimate the body pose. All skeleton points published as TF frames are suffixed with the same `<bodyID>`, thus enabling several unique skeletons to be tracked and visible in TF simultaneously (not visible on Fig. 2 for clarity).

Following the REP-103, the x-axis of the frames points forward (i.e., out of the body), while the z-axis points toward the head (i.e. up when the person is standing vertically, with arm resting along the body).

[6] http://wiki.ros.org/urdf
[7] https://github.com/ros4hri/hri_skeletons
[8] https://github.com/ros4hri/hri_msgs
[9] https://www.ros.org/reps/rep-0103.html
[10] https://www.ros.org/reps/rep-0120.html

TABLE I: Topic structure for human-related signals

/humans/faces/<faceID> (for instance, /humans/faces/bf3d)

Name	Message type	Description
/roi	sensor_msgs/RegionOfInterest	Region of the face in the source image
/landmarks	hri_msgs/FacialLandmarks	The 2D facial landmarks extracted from the face
/facs	hri_msgs/FacialActionUnits	The presence and intensity of facial action units found in the face
/expression	hri_msgs/Expression	The expression recognised from the face

/humans/bodies/<bodyID>

Name	Message type	Description
/roi	sensor_msgs/RegionOfInterest	Region of the whole body in the source image
/skeleton2d	hri_msgs/Skeleton2D	The 2D points of the detected skeleton
/attitude	hri_msgs/BodyAttitude	Recognised body attitude or gesture

(3D skeletons and poses are represented through TF frames)

/humans/voices/<voiceID>

Name	Message type	Description
/audio	audio_msgs/AudioData	Separated audio stream for this voice
/features	hri_msgs/AudioFeatures	INTERSPEECH'09 Emotion challenge [19] low-level audio features.
/is_speaking	std_msgs/Bool	Whether or not speech is recognised from this voice
/speech	std_msgs/String	The live stream of speech recognized via an ASR engine

/humans/persons/<personID>

Name	Message type	Description
/face_id	std_msgs/String (latched)	Face matched to that person (if any)
/body_id	std_msgs/String (latched)	Body matched to that person (if any)
/voice_id	std_msgs/String (latched)	Voice matched to that person (if any)
/location_confidence	std_msgs/Float32	Location confidence; 1 means 'person currently seen', 0 means 'person location unknown'
/demographics	hri_msgs/AgeAndGender	Detected age and gender of the person
/name	std_msgs/String	Name, if known
/native_language	std_msgs/String	IETF language codes like EN_gb, if known

/humans/interactions

Name	Message type	Description
/groups	hri_msgs/GroupsStamped	Estimated social groups
/gaze	hri_msgs/GazesStamped	Estimated gazing behaviours

The 15 links are connected through 18 joints: 3 degrees of freedom (DoF) for the head, 3 DoFs for each shoulder, 1 DoF for elbows and knees, 2 DoFs for the hips, and 1 DoF for the waist. In the current version, the wrists and ankles are not articulated (due to the lack of support for tracking hands and feet in 3D pose estimators), but this could be easily added in future revisions.

2) Face frame: Head pose estimation modules are requested to publish the 6D head pose as a TF frame named face_<faceID> where <faceID> stands for the unique face identifier of this face. The parent of this frame is the sensor frame used to estimate the face pose. The origin of the frame must be the sellion (defined as the deepest midline point of the angle formed between the nose and forehead. It can generally be approximated to the midpoint of the line connecting the two eyes). The x-axis is expected to point forward (i.e., out of the face), the z-axis is expected to point toward the scalp (i.e., up when the person is standing vertically).

Head vs face frames If the skeleton tracker provides an estimate of the head pose, it might publish a frame named head_<bodyID>, located at the sellion (mid-point between the two eyes). It is the joint responsibility of the face tracker and skeleton tracker to ensure that face_<faceID> head_<bodyID> are consistent with each other, e.g. collocated.

Gaze In addition to the face, a head pose estimator might publish a TF frame representing the gaze direction, gaze_<faceID>. While collocated with the face frame, it follows the convention of cameras' optical frames: the z-axis points forward, the y-axis points down.

3) Person frame: The person_<personID> frame has a slightly more complex semantic and needs to be interpreted in conjunction with the value published on the topic /humans/persons/<personID>/location_confidence.

We can distinguish three cases:

- The human is currently being tracked (i.e. personID is set, and at least one of faceID, bodyID or voiceID is set). In this case, location_confidence should be 1, and:
  1) if a face is associated to the person, the person_<personID> frame must be collocated

TABLE II: List of newly introduced ROS messages for HRI

Message name	Motivation
AgeAndGender	As mentioned in [13], age and gender are key demongraphic factors when it comes to user acceptance of robots. The message encode both age and gender, with associated levels of confidence.
AudioFeatures	Encodes 16 low-level audio features, based on the INTERSPEECH'09 Emotion recognition challenge [19].
BodyAttitude	Body posture recognition is essential when designing cooperative robots [9]. The message encodes three such categorical body postures (hands on face, arms crossed, hands raised), and could be easily extended in the future.
Expression	Expressions and basic emotions are extensively discussed in the literature due to the amount of information they infer about human behaviour. The Expression message encode facial expression, either in a categorical manner (Ekman's model [6]), or using the Valence/Arousal continuous plane.
FacialActionUnits	Encodes the intensity and confidence level of detected Facial Action Units, following the coding scheme and nomenclature proposed in [7].
FacialLandmarks	Encodes the 2D coordinates in image space (and confidence) of 67 facial landmarks (including mouth, nose, eyes, and face silhouette).
Group	List of person IDs being detected as forming a social group. The list of all groups is published as a GroupsStamped message.
GazeSenderReceiver	Encodes one person being observed as gazing at another, as a pair of person IDs. The list of all such gazing behaviour at a given time is published as a GazesStamped message.
Skeleton2D	The message encodes the 3D coordinates of 18 skeletal key points.

with the face_<faceID> frame.

2) else, if a body is associated with the person, the person_<personID> frame must be collocated with the skeleton frame the closest to the head.
3) else, the best available approximation of the person's position (for instance, based on sound source localization) should be used.

- The human is not currently seen/heard, but a prior localization is known. In this case, location_confidence must be set to a value < 1 and a person person_<personID> TF frame must be published as long as location_confidence > 0. Simple implementations might choose to publish location_confidence $=$ 0.5 as soon as the person is not actively seen anymore, while continuously broadcasting the last known location. More advanced implementations might slowly decrease location_confidence over time (to represent the fact that the human might have walked away, for instance), eventually stopping to publish the person_<personID> frame.
- The system knows about the person (for instance, from dialogue with another person), but has no location information. In this case, location_confidence must be set to 0, and no TF frame should be broadcast.

V. Reference pipeline

The ROS4HRI framework does not enforce any specific implementation, neither in terms of algorithms/modules to process signals, nor in terms of overall pipeline design. Enabling roboticists to flexibly design or tailor a social signal processing pipeline appropriate for their target application or available sensors and compute capabilities is a key design goal of ROS4HRI. By defining shared interfaces, ROS4HRI simplifies however the modular design of such pipeline, and enable a high level of code sharing and interoperability.

Figure 3 presents however a possible implementation of the ROS4HRI pipeline, using existing mature open-source software packages to implement the various social signal processing tasks. We specifically provide an open-source implementation of a subset of this pipeline (grey area in Figure 3). Our implementation[11] extracts and represents the following features using the ROS4HRI conventions:

- **Facial landmarks**, used in particular to determine the action units. Extracted through OpenFace [2].
- **Facial action units**, can be used as inputs to eg emotions classifiers. Extracted through OpenFace.
- **6D head pose**, can be used to infer proximity between people and approximate gaze direction. Extracted through OpenFace.
- **Gaze direction**, using head pose estimation and pupil detection. Extracted through OpenFace.
- **Age and Gender**, estimated using the OpenVINO[12] *age-gender-recognition* model.
- **2D and 3D skeletal key-points**: 18 body key-points are detected using OpenVINO, both in 2D and in 3D, also supporting multiple people. The 3D key-points are used to generate on-the-fly URDF models of the detected persons, as well as computing their joint state (using the ikpy inverse kinematics library[13]). Automatically-spawned instances of ROS's robot_state_publisher are then responsible for publishing a kinematically consistent TF tree for each person.
- **Body symbolic pose**: the upper body pose is detected by using the distances between the first 7 points detected by the OpenPose COCO model and can classify: hands-on face, hands raised, and arms crossed. All three classifications can infer the degree of engagement of the person in the interaction.

In addition, person tracking and re-identification is provided by OpenVINO. All these features are exposed to the system using the ROS HRI messages and topics presented in the previous section.

[11] https://github.com/ros4hri/ros4hri
[12] https://01.org/openvinotoolkit
[13] https://github.com/Phylliade/ikpy

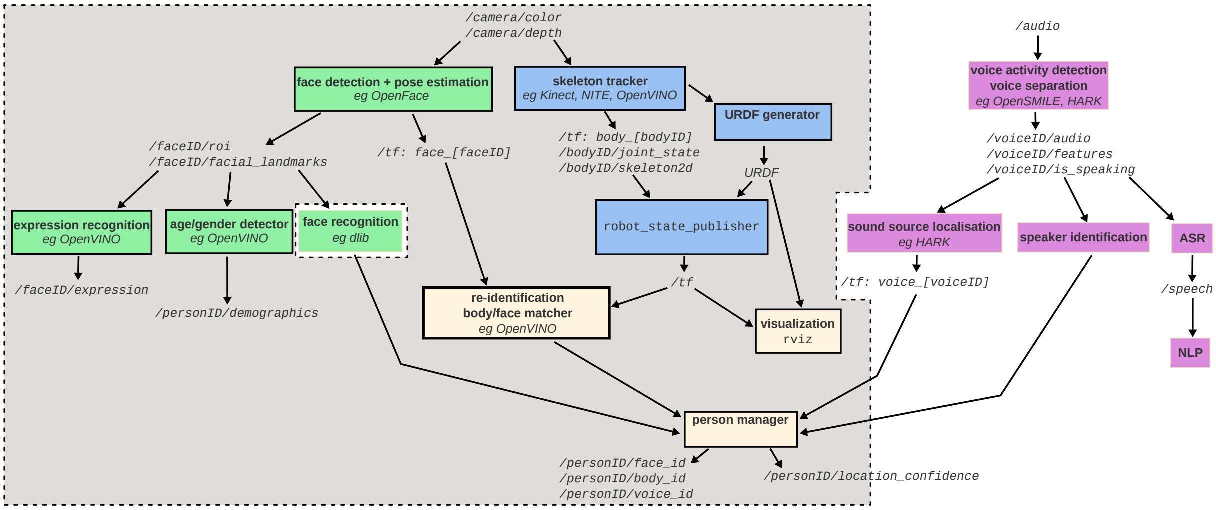

Fig. 3: Reference signal processing pipeline. Green nodes (left) process facial signals, blue nodes (middle) deal with the body tracking, while purple nodes (right) implement the audio processing part. Light yellow nodes, at the bottom, deal with modalities fusion, and manage the permanent `personIDs`. The grey area represents the nodes present in our reference pipeline implementation.

A. Evaluation

We demonstrate and evaluate our pipeline on a small dataset of complex multi-party social interaction based on the MAFIA game. Six participants played during a 20 minute session, with only three visible to the camera (Figure 4).

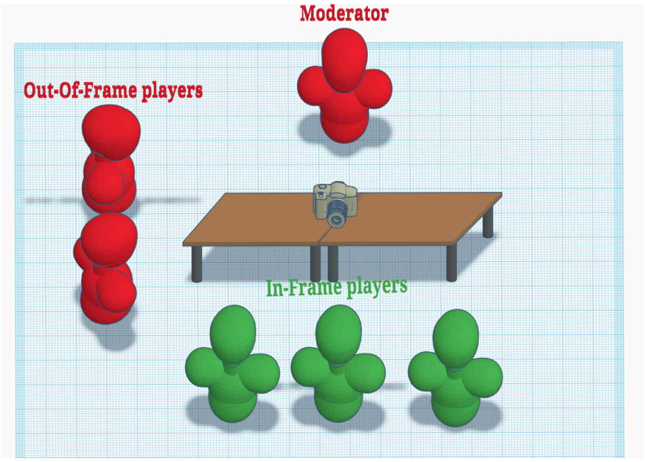

Fig. 4: Game formation

We ran our pipeline on the CPU only (Intel Core i7-6700HQ CPU @ 2.60GHz). While all the expected features were properly detected, the framerate was low (4 FPS) due to the multiple neural nets running in parallel. Running the pipeline on a GPU would significantly improve the performance; however, we have not been able to test it as the OpenVINO models used for the development (OpenVINO 2020.4) did not all support GPU acceleration. This is however expected to improve rapidly.

We specifically evaluated the gaze direction algorithm by comparing the detected gaze instances (as published on `/humans/interactions/gaze` with manual annotations. The gaze detection algorithm correctly estimated gazing behaviours in 78% of the gaze instances.

VI. Future work and conclusion

A. Further integration into the ROS ecosystem

We aim at submitting a *ROS Enhancement Proposal* (REP) to formally specify the ROS4HRI proposal. As such, this article also aims at engaging the community with this design effort, and we want to use the project's public issue tracker to record feedback, and foster further discussion on the proposal, ahead of its submission as a ROS REP.

Our work is currently focused on ROS1 instead of ROS2, mostly due to the extensive amount of code and algorithms available within the ROS1 ecosystem. Once the ROS4HRI design is fully stabilised, we will certainly consider porting it to ROS2. In particular, the messages and topics structure should be straightforwardly transferable.

B. Conclusion

The article presents the *ROS4HRI* framework. ROS4HRI consists of two parts: a model to flexibly represent humans for HRI applications, and a set of ROS datatype and conventions to facilitate the construction of complex multi-modal pipelines for HRI and social signal processing.

Our human model has three important features: (1) it takes into account the different requirements of different HRI applications by modularizing the model into four parts (human body, human face, human voice and human 'person') that can be used independently or together; (2) it takes into account the practicalities of social signal acquisition (like the importance of re-identification) by introducing a system based on unique, transient IDs, that enables a clean separation of concerns between (face, body, voice) detection on one hand, and tracking and fusion on the other hand; (3)

it does not make any assumptions regarding specific tools or packages that could be used in an implementation.

Our ROS specification introduces a small set of new ROS messages (re-using existing ones when sensible); it sets out a set of conventions regarding the structure of HRI-related topics, tightly integrating the unique human IDs into the naming scheme; introduce a kinematic model of the human that implements existing ROS conventions, using dynamically generated URDF models to match the different sizes of each person, while leveraging existing ROS tools for visualization.

Finally, the article introduces a ROS reference pipeline for HRI, as well as a partial open-source implementation of the pipeline (including faces, bodies and persons processing). The pipeline consists of new ROS wrappers around existing software packages like OpenFace and OpenVINO, as well as entirely new nodes, like the dynamic URDF generator or the 'person' manager.

Together, these three contributions (human model, ROS specification, and reference implementation) significantly contribute to close the 'HRI gap' in the ROS ecosystem. This article also aims at engaging the HRI community with this specification effort, and, at the term of this process, we intend to submit a new ROS REP to formally specify our model and conventions.

Acknowledgements

The authors would like to thank the Bristol Robotics Lab colleagues (in particular, Alex Smith and Manuel Giuliani) who contributed to the early design discussions of the ROS4HRI framework.

References

[1] Badino, L., Canevari, C., Fadiga, L., and Metta, G. Integrating articulatory data in deep neural network-based acoustic modeling. *Computer Speech & Language 36* (2016), 173–195.

[2] Baltrusaitis, T., Zadeh, A., Lim, Y. C., and Morency, L. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)* (2018), pp. 59–66.

[3] Bormann, R., Zwölfer, T., Fischer, J., Hampp, J., and Hägele, M. Person recognition for service robotics applications. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (2013), IEEE, pp. 260–267.

[4] Burgoon, J. K., Magnenat-Thalmann, N., Pantic, M., and Vinciarelli, A. *Social signal processing*. Cambridge University Press, 2017.

[5] D'Haro, L. F., Niculescu, A. I., Cai, C., Nair, S., Banchs, R. E., Knoll, A., and Li, H. An integrated framework for multimodal human-robot interaction. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (2017), IEEE, pp. 076–082.

[6] Ekman, P. An argument for basic emotions. *Cognition & emotion 6*, 3-4 (1992), 169–200.

[7] Ekman, P., and Friesen, W. Facial action coding system: A technique for the measurement of facial movement.

[8] Fong, T., Kunz, C., Hiatt, L. M., and Bugajska, M. The human-robot interaction operating system. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction* (2006), pp. 41–48.

[9] Gaschler, A., Jentzsch, S., Giuliani, M., Huth, K., de Ruiter, J., and Knoll, A. Social behavior recognition using body posture and head pose for human-robot interaction. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), IEEE, pp. 2128–2133.

[10] Hebesberger, D., Dondrup, C., Koertner, T., Gisinger, C., and Pripfl, J. Lessons learned from the deployment of a long-term autonomous robot as companion in physical therapy for older adults with dementia: A mixed methods study. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction* (2016), HRI '16, IEEE Press, p. 27–34.

[11] Huang, C.-M., and Mutlu, B. Robot behavior toolkit: generating effective social behaviors for robots. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2012), IEEE, pp. 25–32.

[12] Jang, M., Kim, J., and Ahn, B.-K. A software framework design for social human-robot interaction. In *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)* (2015), IEEE, pp. 411–412.

[13] Kuo, I. H., Rabindran, J. M., Broadbent, E., Lee, Y. I., Kerse, N., Stafford, R., and MacDonald, B. A. Age and gender factors in user acceptance of healthcare robots. In *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication* (2009), IEEE, pp. 214–219.

[14] Lane, I., Prasad, V., Sinha, G., Umuhoza, A., Luo, S., Chandrashekaran, A., and Raux, A. HRItk: The human-robot interaction ToolKit rapid development of speech-centric interactive systems in ROS. In *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012)* (Montréal, Canada, June 2012), Association for Computational Linguistics, pp. 41–44.

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

[16] Li, R., Oskoei, M. A., and Hu, H. Towards ros based multi-robot architecture for ambient assisted living. In *2013 IEEE International Conference on Systems, Man, and Cybernetics* (2013), IEEE, pp. 3458–3463.

[17] Pandey, A. K., and Gelin, R. A mass-produced sociable humanoid robot: Pepper: The first machine of its kind. *IEEE Robotics & Automation Magazine 25*, 3 (2018), 40–48.

[18] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. Ros: an open-source robot operating system. In *ICRA workshop on open source software* (2009), vol. 3, Kobe, Japan, p. 5.

[19] Schuller, B., Steidl, S., and Batliner, A. The interspeech 2009 emotion challenge. In *Tenth Annual Conference of the International Speech Communication Association* (2009).

[20] Twomey, K. E., Morse, A. F., Cangelosi, A., and Horst, J. S. Children's referent selection and word learning. *Interaction Studies 17*, 1 (Sep 2016), 101–127.

[21] Wagner, J., Lingenfelser, F., Baur, T., Damian, I., Kistler, F., and André, E. The social signal interpretation (ssi) framework: multimodal signal processing and recognition in real-time. In *Proceedings of the 21st ACM international conference on Multimedia* (2013), pp. 831–834.

[22] Zhang, Y., and Xu, S. C. Ros based voice-control navigation of intelligent wheelchair. In *Applied Mechanics and Materials* (2015), vol. 733, Trans Tech Publ, pp. 740–744.