Deliverable D7.5

Final software architecture for SPRING-ARI

Due Date: March 2024
Main Author: PAL
Contributors: N/A
Dissemination: PU - Public

This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.
This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

**DOCUMENT FACTSHEET**

<table>
<thead>
<tr>
<th>Deliverable no.</th>
<th>D7.5: Final software architecture for SPRING-ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Partner</td>
<td>PAL</td>
</tr>
<tr>
<td>Work Package</td>
<td>WP7: WP Robot Customization and Software Integration</td>
</tr>
<tr>
<td>Task</td>
<td>T7.5: Final Software Integration Cycle</td>
</tr>
<tr>
<td>Version &amp; Date</td>
<td></td>
</tr>
<tr>
<td>Dissemination level</td>
<td>[X] PU (public) [ ] CO (confidential)</td>
</tr>
</tbody>
</table>

**CONTRIBUTORS AND HISTORY**

<table>
<thead>
<tr>
<th>Version</th>
<th>Editor</th>
<th>Date</th>
<th>Change Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>PAL</td>
<td>05/04/2024</td>
<td></td>
</tr>
<tr>
<td>Final</td>
<td>PAL</td>
<td>10/04/2024</td>
<td>Updated architecture diagrams with latest updates from partners; proof-reading</td>
</tr>
</tbody>
</table>

**APPROVALS**

| Authors/editors | Severin Lemaignan, PAL |
# TABLE OF CONTENTS

Executive Summary ........................................................................................................... 5

Contents of Deliverable .................................................................................................... 6

1. General Overview ......................................................................................................... 7
   Development methodology .............................................................................................. 7
   Dissemination .................................................................................................................. 8
   Progress of the architecture over the course of the project ......................................... 8
   Code availability ............................................................................................................. 10

2. Functional domains ...................................................................................................... 11
   Speech processing .......................................................................................................... 11
   Human localisation ........................................................................................................ 11
   Behaviour analysis ........................................................................................................ 12
   Person modeling ............................................................................................................ 13
   Self-localisation ............................................................................................................. 13
   Multi-party conversation ............................................................................................... 14
   Non-verbal behaviours ................................................................................................. 15
   User interfaces ............................................................................................................. 15

3. Complete list of ROS nodes ....................................................................................... 16

4. Conclusion ................................................................................................................... 20

5. Appendix: Detailed Description of All Nodes ........................................................... 21
   BIU ............................................................................................................................... 21
   CVUT ............................................................................................................................ 24
   ERM ............................................................................................................................. 26
   HWU ............................................................................................................................. 27
EXECUTIVE SUMMARY

The final architecture of the SPRING robot comprises of 52 ROS nodes, organised in 8 functional domains, and more than 170 ROS topics, services, and actions connecting these nodes together.

The architecture is designed to be modular, scalable, and flexible, to allow for easy integration of new components, and to facilitate the development of new behaviours and functionalities.

The development of the SPRING architecture has been carried out in an iterative manner, with regular integration and testing cycles, and has followed the principles of the ROS framework and the ROS4HRI standard.

The architecture has been presented at several conferences and workshops, and a consortium-wide journal submission describing the architecture and its components is currently under review at the International Journal of Social Robotics.

The final architecture of the SPRING robot has evolved significantly over the course of the project, and is now implemented and deployed on the PAL ARI robot.

The architecture specification is available at https://gitlab.inria.fr/spring/wp7_ari/spring-architecture while the open-source implementation of the main ROS nodes is publicly available from https://gitlab.inria.fr/spring/.
CONTENTS OF DELIVERABLE

The aim of this deliverable is to present the final software architecture of the SPRING robot, successively defined and iteratively refined in tasks T7.3 (Preliminary Software Integration Cycle), T7.4 (Intermediate Software Integration Cycle) and T7.5 (Final Software Integration Cycle).

It presents the resulting Final Software Architecture of the SPRING ARI robot, and lists the different software modules present in the architecture.

As the Deliverable is a Code Deliverable, the textual description of the architecture is relatively concise. The code itself can be found on the project Git forge:

- architecture specification: https://gitlab.inria.fr/spRING/wp7_ari/spring-architecture
- main ROS nodes: https://gitlab.inria.fr/spRING/

AI transparency disclaimer: parts of this report have been written with the help of an AI language model. These parts have been carefully checked for accuracy by the authors, who take full responsibility for the entirety of the report.
1. GENERAL OVERVIEW

The aim of this deliverable is to present the final software architecture of the SPRING robot, successively defined and iteratively refined in tasks T7.3 (Preliminary Software Integration Cycle), T7.4 (Intermediate Software Integration Cycle) and T7.5 (Final Software Integration Cycle).

The purpose of these tasks is to define the software architecture of the SPRING, and to integrate the software components developed by the SPRING partners, to achieve a fully autonomous robot capable of performing the tasks and scenarios defined in WP1.

The final architecture comprises of:

- 52 SPRING-specific ROS nodes, organised in 8 functional domains (speech processing, behaviour analysis, human localisation, person modeling, self-localisation, multi-party conversation, non-verbal behaviours, user interfaces). (note that these nodes communicate with the functional layers of the PAL ARI robot, which themselves are composed of many additional ROS nodes);

- more than 170 ROS topics, services, and actions connecting these nodes together.

Figure 1 shows an overview of the final architecture of the SPRING robot, with the 8 domains.

![Figure 1: Overview of the final architecture of the SPRING robot](image-url)
The architecture is designed to be modular, scalable, and flexible, to allow for easy integration of new components, and to facilitate the development of new behaviours and functionalities.

In particular, we follow the principles of the ROS framework, which is widely used in the robotics community, and provides a standardised way to develop and integrate software components. The SPRING project has actively engaged with the ROS community and has contributed the development of the new ROS standard REP-155, so-called ROS4HRI. This standard defines a set of guidelines and best practices for developing ROS nodes for human-robot interaction and aims to facilitate the development of HRI systems by providing a common framework and set of tools.

Not only does the SPRING architecture follow the ROS4HRI guidelines, but PAL Robotics has also adopted this standard for the range of robots that they manufacture and sell, a major exploitation outcome of the SPRING project.

The development of the SPRING architecture has been carried out in an iterative manner, with regular integration and testing cycles (four integration weeks over the course of the project, as well as long-term deployment at the Broca hospital).

We have used and adapted the Boxology tool to visualise the architecture, and to facilitate the communication between the partners.

The tool has been extended to automatically generate ROS node skeletons and documentation. In particular, the detailed list of nodes in the SPRING architecture, appended to this deliverable, has been generated using this tool.

**Dissemination**

The various parts of the SPRING architecture have been presented at several conferences and workshops, and a consortium-wide journal publication describing at high-level the architecture and its components is currently under review at the International Journal of Social Robotics.

**Progress of the architecture over the course of the project**

Figure 2 presents the SPRING architecture, as it was initially defined in the deliverable D7.3 (Initial Software Architecture).
This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Figure 2: Initial architecture of the SPRING robot, as presented in the deliverable D7.3

When comparing it to Figure 1, we can see that the final architecture is more focused on functional domains (socio-cognitive capabilities), and less on the project’s rigid workpackage structure. This reflects the evolution of the project, and the acknowledgment that many of the robot’s capabilities require in fact a combination of the expertise of several partners.

Figure 3 presents the final architecture of the SPRING robot, as currently implemented and deployed on the PAL ARI robot. The individual contributions of the partners are still visible in this diagram, color-coded by partner (see Figure 4).

Figure 3: Final architecture of the SPRING robot. We ‘zoom in’ on each domain below.
Also visible in Figure 3 is the significant increase in the number of nodes and interfaces, compared to the initial architecture. This reflects both the actual complexity of the system, as clarified by the integration work, and a better understanding of the mechanisms and interfaces required to achieve a modular system, as developed by all the partners over the course of the project.

Code availability

This deliverable is a code deliverable. As specified at the beginning of the project, the code itself is hosted as a set of git repositories at https://gitlab.inria.fr/spring.

In particular, the SPRING architecture is formally described in this repository: https://gitlab.inria.fr/spring/wp7_ari/spring-architecture.

The list of ROS nodes and their relationships, as presented below, has been automatically generated from the formal architecture description using the Boxology tool, as previously explained.
2. FUNCTIONAL DOMAINS

Speech processing

Speech processing domain

This domain is responsible for processing speech input from the user, including audio enhancement and speech recognition. The work has mostly been carried out in WP3 by BIU.

It is composed of the following nodes:

- riva_asr
- speaker_separation
- CSD
- speaker_extraction
- single_speaker_noise_reduction
- speakers_id_and_DOA
- audio Arbitr

Human localisation

Human localisation domain

This domain is responsible for localising humans in the environment, including 2d/3D body pose estimation, and face detection. The work has mostly been carried out in WP3 and WP4 by INRIA and UNITN.
It is composed of the following nodes:

- `front_fisheye_body_tracker`
- `front_fisheye_2d_body_pose_detector`
- `body_3d_tracker`
- `face_tracker`

**Behaviour analysis**

This domain is responsible for analysing the behaviour of the users, including softbiometrics recognition, emotion estimation, and gaze estimation. The work has mostly been carried out in WP4 by UNITN.

It is composed of the following nodes:

- `soft_biometrics_estimator`
- `mask_detector`
- `human_2d_pose_estimation`
- `emotion_estimation`
- `gaze_estimation`
Person modeling domain

This domain is responsible for modelling the users, including voice-body and face-body matching and tracking. The work has mostly been carried out in WP7 by PAL.

It is composed of the following nodes:

- person_manager
- voicebodymatching
- body_to_face_mapper

Self-localisation domain

D7.5 Final software architecture for SPRING-ARI | |Page 13
This domain is responsible for localising the robot in the environment, including SLAM and global localisation. The work has mostly been carried out in WP2 by CVUT.

It is composed of the following nodes:

- HLoc
- slam_rtabmap

**Multi-party conversation domain**

This domain is responsible for managing multi-party conversations, including dialogue management and social state analysis. The work has mostly been carried out in WP5 by HWU.

It is composed of the following nodes:

- dialogue_speech
- dialogue_arbitrer
- dialogue_nlp
- dialogue_say
- knowledge_core
- social_state_analyzer
- social_strategy_supervisor
- recipe_planner
- ros_petri_net_node
- interaction_manager
Non-verbal behaviours

This domain is responsible for generating non-verbal behaviours, including socially-aware navigation, and gaze control. The work has mostly been carried out in WP6 by INRIA.

It is composed of the following nodes:

- **behavior_generator**
- **multiple look_at_*_action_server**
- **multiple go_to_*_action_server**

User interfaces

This domain is responsible for providing user interfaces, including the robot tablet interface and the experimenter tablet. The work has mostly been carried out in WP1 by ERM.

It is composed of the following nodes:

- **robotgui**
- **experimenter_gui**
- **experiment_exporter**
### 3. COMPLETE LIST OF ROS NODES

<table>
<thead>
<tr>
<th>Node</th>
<th>Partner</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>riva_asr</td>
<td>BIU</td>
<td>released</td>
<td>Speech recognition based on NVIDIA Riva (2nd instance for 2nd speaker). Code: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/riva_asr_ros_client">https://gitlab.inria.fr/spring/wp3_av_perception/riva_asr_ros_client</a></td>
</tr>
<tr>
<td>speaker_separation</td>
<td>BIU</td>
<td>released</td>
<td>blind separation + VAD + enhancement. Code: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/audio_separation.git">https://gitlab.inria.fr/spring/wp3_av_perception/audio_separation.git</a></td>
</tr>
<tr>
<td>CSD</td>
<td>BIU</td>
<td>not yet implemented</td>
<td>Voice embedding + speaker separation together. Code: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/1ch_speaker_extraction">https://gitlab.inria.fr/spring/wp3_av_perception/1ch_speaker_extraction</a></td>
</tr>
<tr>
<td>speaker_extraction</td>
<td>BIU</td>
<td>released</td>
<td>Voice embedding + speaker separation together. Code: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/1ch_speaker_extraction">https://gitlab.inria.fr/spring/wp3_av_perception/1ch_speaker_extraction</a></td>
</tr>
<tr>
<td>single_speaker_noise_reduction</td>
<td>BIU</td>
<td>released</td>
<td>Audio pre-processing (incl. noise cancellation). Code: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/speech-enhancement">https://gitlab.inria.fr/spring/wp3_av_perception/speech-enhancement</a></td>
</tr>
<tr>
<td>speakers_id_and_DOA</td>
<td>BIU</td>
<td>released</td>
<td>Speaker identification based on voice embeddings. Code: <a href="https://gitlab.inria.fr/spring/wp4_behavior/non-integrated-contributions/speaker_identification">https://gitlab.inria.fr/spring/wp4_behavior/non-integrated-contributions/speaker_identification</a></td>
</tr>
<tr>
<td>audio_arbitrer</td>
<td>BIU</td>
<td>released</td>
<td>Assigns voices and audio stream, and publishes ROS4HRI-compatible ROS messages. Code: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/audio_manager">https://gitlab.inria.fr/spring/wp3_av_perception/audio_manager</a></td>
</tr>
<tr>
<td>slam_rtabmap</td>
<td>CVUT</td>
<td>released</td>
<td>RTABmap based SLAM. Code: <a href="https://gitlab.inria.fr/spring/wp2_mapping_localization/rtabmap-orbslam2">https://gitlab.inria.fr/spring/wp2_mapping_localization/rtabmap-orbslam2</a></td>
</tr>
<tr>
<td>HLoc</td>
<td>CVUT</td>
<td>released</td>
<td>Global localization, service-based. Code: <a href="https://gitlab.inria.fr/spring/wp2_mapping_localization/hloc-mapping-localization.git">https://gitlab.inria.fr/spring/wp2_mapping_localization/hloc-mapping-localization.git</a></td>
</tr>
<tr>
<td>Yolact3D</td>
<td>CVUT</td>
<td>released</td>
<td>Object detection/identification/localisation. Code: <a href="https://gitlab.inria.fr/spring/wp2_mapping_localization/yolact3d.git">https://gitlab.inria.fr/spring/wp2_mapping_localization/yolact3d.git</a></td>
</tr>
<tr>
<td>experiment_exporter</td>
<td>ERM</td>
<td>released</td>
<td>Experiment_exporter is in charge of logging the experiment data. Code: <a href="https://gitlab.inria.fr/spring/wp1_user_application/">https://gitlab.inria.fr/spring/wp1_user_application/</a></td>
</tr>
<tr>
<td>Component</td>
<td>Type</td>
<td>Status</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>robot gui</td>
<td>ERM</td>
<td>released</td>
<td>Robot tablet interface. Code:</td>
</tr>
<tr>
<td>experimenter_gui</td>
<td>ERM</td>
<td>released</td>
<td>Web server for the experimenter tablet. Code:</td>
</tr>
<tr>
<td>dialogue_say</td>
<td>HWU</td>
<td>released</td>
<td>proxy to robot's TTS action server for ERM convenience. Code:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUBFOLDER:dialogue_say</td>
</tr>
<tr>
<td>interaction_manager</td>
<td>HWU</td>
<td>released</td>
<td>Code:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUBFOLDER:interaction_manager</td>
</tr>
<tr>
<td>ros_petri_net_node</td>
<td>HWU</td>
<td>released</td>
<td>Petrinet-based task planning. Code:</td>
</tr>
<tr>
<td>dialogue_speech</td>
<td>HWU</td>
<td>released</td>
<td>Speech pre-processing (incl. end of speech detection). Code:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUBFOLDER:dialogue_speech</td>
</tr>
<tr>
<td>nlp_node</td>
<td>HWU</td>
<td>released</td>
<td>ALANA chatbot</td>
</tr>
<tr>
<td>dialogue_arbitrer</td>
<td>HWU</td>
<td>released</td>
<td>Dialogue arbitrer. Code:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUBFOLDER:dialogue_arbitrer</td>
</tr>
<tr>
<td>recipe_planner</td>
<td>HWU</td>
<td>released</td>
<td>Code:</td>
</tr>
<tr>
<td>social_state_analyzer</td>
<td>HWU</td>
<td>released</td>
<td>Social scene understanding</td>
</tr>
<tr>
<td>social_strategy_supervisor</td>
<td>HWU</td>
<td>released</td>
<td>High-level interaction supervisor</td>
</tr>
<tr>
<td>go_to_group_action_server</td>
<td>INRIA</td>
<td>released</td>
<td>Robot action server (group approach)</td>
</tr>
<tr>
<td>go_to_person_action_server</td>
<td>INRIA</td>
<td>released</td>
<td>Robot action server (person approach)</td>
</tr>
<tr>
<td>occupancy_map_publisher</td>
<td>INRIA</td>
<td>not yet implemented</td>
<td></td>
</tr>
<tr>
<td>front_fisheye_2d_body_pose_detector_op</td>
<td>INRIA</td>
<td>released</td>
<td>2D skeleton estimator This node estimates the 2.5D (x,y,theta) pose of nearby persons.. Code:</td>
</tr>
<tr>
<td>Component Name</td>
<td>Organization</td>
<td>Status</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>group_detector</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Group detection (incl. f-formations). Code: <a href="https://gitlab.inria.fr/spring/wp4_behavior/group_detector">https://gitlab.inria.fr/spring/wp4_behavior/group_detector</a></td>
</tr>
<tr>
<td><strong>body_to_face_mapper</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Face-body matching. Code: node associates detected bodies to detected faces in image-space. REPO: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/body_to_face_mapper">https://gitlab.inria.fr/spring/wp3_av_perception/body_to_face_mapper</a></td>
</tr>
<tr>
<td><strong>basestation_republisher</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Node to republish compressed video streams on the SPRING basestation. Code: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/docker_republisher">https://gitlab.inria.fr/spring/wp3_av_perception/docker_republisher</a></td>
</tr>
<tr>
<td><strong>go_to_body_action_server</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Robot action server (body approach)</td>
</tr>
<tr>
<td><strong>go_to_position_action_server</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Robot action server (navigation to location)</td>
</tr>
<tr>
<td><strong>look_at_person_server</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Robot action server (‘look at person’ server)</td>
</tr>
<tr>
<td><strong>look_at_action_server</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Robot action server (generic ‘look at’ action)</td>
</tr>
<tr>
<td><strong>social_mpc</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Code: <a href="https://gitlab.inria.fr/spring/wp6_robot_behavior/robot_behavior">https://gitlab.inria.fr/spring/wp6_robot_behavior/robot_behavior</a></td>
</tr>
<tr>
<td><strong>look_at_position_action_server</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Robot action server (‘look at location’ action)</td>
</tr>
<tr>
<td><strong>pointcloud2occmap</strong></td>
<td>INRIA</td>
<td>released</td>
<td>2D body detector and tracker, based on the FairMOT algorithm. Code: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/multi-person_visual_tracker/">https://gitlab.inria.fr/spring/wp3_av_perception/multi-person_visual_tracker/</a></td>
</tr>
<tr>
<td><strong>front_fisheye_2d_body_tracker</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Matching between localised voices and detected bodies</td>
</tr>
<tr>
<td><strong>voicebodycandidatematches</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Laser-based 3D people pose estimation. REPO: <a href="https://gitlab.inria.fr/spring/wp3_av_perception/body_3d_tracker">https://gitlab.inria.fr/spring/wp3_av_perception/body_3d_tracker</a></td>
</tr>
<tr>
<td><strong>body_3d_tracker</strong></td>
<td>INRIA</td>
<td>released</td>
<td>Microphone array driver. REPO: <a href="https://gitlab.inria.fr/spring/wp7_ari/respeaker_ros">https://gitlab.inria.fr/spring/wp7_ari/respeaker_ros</a></td>
</tr>
<tr>
<td><strong>respeaker_ros</strong></td>
<td>PAL</td>
<td>released</td>
<td>RGB head camera driver</td>
</tr>
<tr>
<td><strong>fisheye</strong></td>
<td>PAL</td>
<td>released</td>
<td>RGB head camera driver</td>
</tr>
<tr>
<td><strong>raspicam</strong></td>
<td>PAL</td>
<td>released</td>
<td>RGB head camera driver</td>
</tr>
<tr>
<td>Component</td>
<td>Provider</td>
<td>Status</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Robot functional layer</strong></td>
<td><strong>PAL</strong></td>
<td>released</td>
<td>robot's hardware interfaces</td>
</tr>
<tr>
<td><strong>hri_person_manager</strong></td>
<td><strong>PAL</strong></td>
<td>released</td>
<td>Probabilistic fusion of faces, bodies, voices into persons. Code: <a href="https://gitlab.inria.fr/spring/wp7_ari/hri_person_manager.git">https://gitlab.inria.fr/spring/wp7_ari/hri_person_manager.git</a></td>
</tr>
<tr>
<td><strong>torso_rgbd_camera</strong></td>
<td><strong>PAL</strong></td>
<td>released</td>
<td>Frontal RGB-D camera driver</td>
</tr>
<tr>
<td><strong>people_facts</strong></td>
<td><strong>PAL</strong></td>
<td>released</td>
<td>Semantic bridge between human perception and the knowledge base. Code: <a href="https://gitlab.inria.fr/spring/wp7_ari/people_facts">https://gitlab.inria.fr/spring/wp7_ari/people_facts</a></td>
</tr>
<tr>
<td><strong>knowledge_core</strong></td>
<td><strong>PAL</strong></td>
<td>released</td>
<td>Robot's RDF/OWL knowledge base. Code: <a href="https://gitlab.inria.fr/spring/wp7_ari/knowledge_core">https://gitlab.inria.fr/spring/wp7_ari/knowledge_core</a></td>
</tr>
<tr>
<td><strong>soft_biometrics_estimator</strong></td>
<td><strong>UNITN</strong></td>
<td>released</td>
<td>Detects age and gender. Code: <a href="https://gitlab.inria.fr/spring/wp4_behavior/wp4_behavior_understanding">https://gitlab.inria.fr/spring/wp4_behavior/wp4_behavior_understanding</a></td>
</tr>
<tr>
<td><strong>emotion_estimation</strong></td>
<td><strong>UNITN</strong></td>
<td>released</td>
<td>Non-verbal behaviour generation</td>
</tr>
<tr>
<td><strong>face_tracker</strong></td>
<td><strong>UNITN</strong></td>
<td>released</td>
<td>face detection and tracking. Code: <a href="https://gitlab.inria.fr/spring/wp4_behavior/face-tracker">https://gitlab.inria.fr/spring/wp4_behavior/face-tracker</a></td>
</tr>
<tr>
<td><strong>mask_detector</strong></td>
<td><strong>UNITN</strong></td>
<td>released</td>
<td>Detects presence of a facial mask. Code: <a href="https://gitlab.inria.fr/spring/wp4_behavior/mask-detection">https://gitlab.inria.fr/spring/wp4_behavior/mask-detection</a></td>
</tr>
<tr>
<td><strong>depth_estimation</strong></td>
<td><strong>UNITN</strong></td>
<td>released</td>
<td>Monocular depth estimation. Code: <a href="https://gitlab.inria.fr/spring/wp4_behavior/depth-estimation">https://gitlab.inria.fr/spring/wp4_behavior/depth-estimation</a></td>
</tr>
<tr>
<td><strong>gaze_estimation</strong></td>
<td><strong>UNITN</strong></td>
<td>released</td>
<td>Monocular gaze estimation on planar image This node uses deep learning to estimate, on a given frame, the focus of attention of a detected face. It outputs the 2D coordinate of the most likely focus of attention, in the image space. REPO: <a href="https://gitlab.inria.fr/spring/wp4_behavior/gaze-estimation">https://gitlab.inria.fr/spring/wp4_behavior/gaze-estimation</a></td>
</tr>
<tr>
<td><strong>activity_recognition</strong></td>
<td><strong>UNITN</strong></td>
<td>released</td>
<td>Activity recognition</td>
</tr>
</tbody>
</table>
4. CONCLUSION

The final architecture of the SPRING robot comprises of 52 ROS nodes, organised in 8 functional domains, and more than 170 ROS topics, services, and actions connecting these nodes together.

The architecture is designed to be modular, scalable, and flexible, to allow for easy integration of new components, and to facilitate the development of new behaviours and functionalities.

The development of the SPRING architecture has been carried out in an iterative manner, with regular integration and testing cycles, and has followed the principles of the ROS framework and the ROS4HRI standard.

The architecture has been presented at several conferences and workshops, and a consortium-wide journal publication describing the architecture and its components is currently under review at the International Journal of Social Robotics.
5. APPENDIX: DETAILED DESCRIPTION OF ALL NODES

BIU

audio_arbitrer

The node audio_arbitrer (id: audio_arbitrer) is maintained by BIU.

Status
Implemented. Current release/branch: 0.0.1
Source code repository: https://gitlab.inria.fr/spring/wp3_av_perception/audio_manager

Inputs
Input: count_active_speakers
Input: processed_audio
Input: active_voices

Outputs
Output: tf: /voice_* (tf)

Topic publication: /humans/voices/<id>/speech [hri_msgs/LiveSpeech]
Topic publication: /humans/voices/<id>/doa [std_msgs/Float32]

Dependencies
std_msgs/Empty
tf/transform_broadcaster
hri_msgs/LiveSpeech
std_msgs/Float32

CSD

The node CSD (id: csd) is maintained by BIU.

Status
This node is not yet implemented.

Inputs
Input: /audio/raw_audio

Outputs
Output: count_active_speakers

Dependencies
riva_asr

The node riva_asr (id: riva_asr) is maintained by BIU.
Status
Implemented. Current release/branch: main
Source code repository: https://gitlab.inria.fr/spring/wp3_av_perception/riva_asr_ros_client
Inputs
Input: audio

Outputs
Output: text

Dependencies
std_msgs/Empty

single_speaker_noise_reduction

This node performs:
- speech echo cancelation,
- single microphone audio enhancement
Used for the single-speaker only pipeline.
The node single_speaker_noise_reduction (id: single_speaker_noise_reduction) is maintained by BIU.
Status
Implemented. Current release/branch: BIU_dev
Source code repository: https://gitlab.inria.fr/spring/wp3_av_perception/speech-enhancement SUBFOLDER:audio_processing
Inputs
Input: /audio/raw_audio

Outputs
Topic publication: /audio/enh_audio [spring_msgs/RawAudioData]

Dependencies
std_msgs/Empty

spring_msgs/RawAudioData

speaker_extraction

The node speaker_extraction (id: speaker_extraction) is maintained by BIU.
Status
Implemented. Current release/branch: 0.0.1
Source code repository: https://gitlab.inria.fr/spring/wp3_av_perception/1ch_speaker_extraction
**speaker_separation**

The node `speaker_separation` (id: speaker_separation) is maintained by BIU.

Status
Implemented. Current release/branch: main

Source code repository:

https://gitlab.inria.fr/spring/wp3_av_perception/audio_separation.git

Inputs
Input: /audio/raw_audio

Outputs
Output: streams

Dependencies
std_msgs/Empty

audio_common_msgs/AudioData

**speakers_id_and_DOA**

The node `speakers_id_and_DOA` (id: speakers_id_and_doa) is maintained by BIU.

Status
Implemented. Current release/branch: dual_speaker_ecapa

Source code repository:

https://gitlab.inria.fr/spring/wp4_behavior/non-integrated-contributions/speaker_identification

Inputs
Input: /audio/voice_stream*

Input: count_active_speakers

Outputs
Topic publication: /humans/voices/<id>/doa [std_msgs/Float32]

Topic publication: /humans/voices/<id>/audio [audio_common_msgs/AudioData]

Dependencies
Docker not yet published

The node HLoc (id: hloc) is maintained by CVUT.

Status
 Implemented. Current release/branch: main

Source code repository: https://gitlab.inria.fr/spring/wp2_mapping_localization/hloc-mapping-localization.git

Inputs
Input: tf: camera frames (tf)

Topic subscription: /robot_pose [geometry_msgs/PoseWithCovarianceStamped]

Input: /front_camera/fisheye/image_raw/compressed
Input: /rear_camera/fisheye/image_raw/compressed

Outputs
Output: ROS service: pose + covariance

Dependencies
tf/transform_listener

std_msgs/Empty

geometry_msgs/PoseWithCovarianceStamped

slam_rtabmap

The node slam_rtabmap (id: slam_rtabmap) is maintained by CVUT.

Status
 Implemented. Current release/branch: 0.0.1

Source code repository: https://gitlab.inria.fr/spring/wp2_mapping_localization/rtabmap-orbslam2

Inputs
Input: torso_front_camera/infra/

Input: /torso_front_camera/imu

Input: [call HLoc to perform global localization]

Outputs
Output: tf: /odom (tf)

Topic publication: /slam/occupancy_map [OccupancyGrid/OccupancyGrid]

Topic publication: /robot_pose [geometry_msgs/PoseWithCovarianceStamped]

Dependencies
tf/transform_broadcaster
std_msgs/Empty
OccupancyGrid/OccupancyGrid
geometry_msgs/PoseWithCovarianceStamped

Yolact3D

Publishes a set of 3D points with the probability distribution of object classes at that point.
ETA: not clear yet.
The node Yolact3D (id: yolact3d) is maintained by CVUT.

Status
Implemented. Current release/branch: master
Source code repository:
https://gitlab.inria.fr/spring/wp2_mapping_localization/yolact3d.git

Inputs
Topic subscription: /slam/occupancy_map [OccupancyGrid/OccupancyGrid]

Input: tf

Input: /torso_front_camera/aligned_depth_to_color [sensor_msgs/Image]

Topic subscription: /torso_front_camera/color/image_raw [sensor_msgs/Image]

Outputs
Topic publication: /yolact3d/detected_objects_distribution [yolact3d/Yolact3DObjects]

Dependencies
yolact3d/Yolact3DObjects
OccupancyGrid/OccupancyGrid

std_msgs/Empty

sensor_msgs/Image

ERM

experiment_exporter

The node `experiment_exporter` (id: `experiment_exporter`) is maintained by ERM.

Status
 Implemented. Current release/branch: main
Source code repository: [https://gitlab.inria.fr/spring/wp1_user_application/export-dialog.git](https://gitlab.inria.fr/spring/wp1_user_application/export-dialog.git)

Inputs

- Topic subscription: `/dialogue_say/text` [String/String]
- Topic subscription: `/experiment_exporter/file_update` [String/String]
- Topic subscription: `/dialogue_speech/eos` [dialogue_msgs/EndOfSpeech]
- Topic subscription: `/experiment_exporter/error_feedback` [String/String]

Outputs

Dependecies

String/String

dialogue_msgs/EndOfSpeech

experimenter_gui

The node `experimenter_gui` (id: `experimenter_gui`) is maintained by ERM.

Status
 Implemented. Current release/branch: main
Source code repository: [https://gitlab.inria.fr/spring/wp1_user_application/exp-gui.git](https://gitlab.inria.fr/spring/wp1_user_application/exp-gui.git)

Inputs

- Input: `/dialogue_start/status`
- Input: `/diagnostics`
- Input: `/slam/localization_pose`

Outputs

Output: `/dialogue_start/goal` [dialogue_msgs/StartDialogueActionGoal]
This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

### robot gui

*The node robot gui (id: robotgui) is maintained by ERM.*

**Status**

Implemented. Current release/branch: master

**Source code repository:**

[https://gitlab.inria.fr/spring/wp1_user_application/user_application.git](https://gitlab.inria.fr/spring/wp1_user_application/user_application.git)

**Inputs**

- Input: /tts/feedback
- Input: /human_dialogue
- Input: /audio/is_listening

**Outputs**

**Dependencies**

- std_msgs/Empty
- Empty/Empty
- pal_web_msgs/WebGoTo
- String/String

### HWU
dialogue_arbitrer

*The node dialogue_arbitrer (id: dialogue_arbitrer) is maintained by HWU.*

**Status**

Implemented. Current release/branch: main
This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Source code repository: https://gitlab.inria.fr/spring/wp5_spoken_conversations/dialogue

**SUBFOLDER:** dialogue_arbiter

**Inputs**

Topic subscription: /dialogue_start [action server/action server]

Topic subscription: /dialogue_arbitrer/end_conv [service/service]

Topic subscription: /dialogue_speech/eos [dialogue_msgs/EndOfSpeech]

**Outputs**

Topic publication: /nlp_node/get_answer [service/service]

Output: /dialogue_start/status

Output: /dialogue_start/feedback

Output: /human_dialogue

Topic publication: /dialogue_say/say [service/service]

Topic publication: /RPN [action/action]

Topic publication: /task*_ros_server_action [action/action]

**Dependencies**

service/service

action server/action server

std_msgs/Empty

action/action

dialogue_msgs/EndOfSpeech

dialogue_say

*The node dialogue_say (id: dialogue_say) is maintained by HWU.*

**Status**

Implemented. Current release/branch: main

Source code repository: https://gitlab.inria.fr/spring/wp5_spoken_conversations/dialogue

**SUBFOLDER:** dialogue_say

**Inputs**

Topic subscription: /dialogue_say/say [service/service]

Input: /speech/speed
Input: /tts/status

Outputs
Topic publication: /dialogue_say/text [String/String]

Output: /tts/feedback

Topic publication: /tts [action_server/action_server]

Dependencies
String/String
service/service
std_msgs/Empty
action_server/action_server

dialogue_speech

The node dialogue_speech (id: dialogue_speech) is maintained by HWU.

Status
Implemented. Current release/branch: master
Source code repository: https://gitlab.inria.fr/spring/wp5_spoken_conversations/dialogue
SUBFOLDER:dialogue_speech

Inputs
Topic subscription: /humans/voices/<id>/speech [hri_msgs/LiveSpeech]

Topic subscription: /social_state_analyser/state [std_msgs/String]

Outputs
Topic publication: /dialogue_speech/eos [dialogue_msgs/EndOfSpeech]

Dependencies
hri_msgs/LiveSpeech
dialogue_msgs/EndOfSpeech
std_msgs/String

interaction_manager

The node interaction_manager (id: interaction_manager) is maintained by HWU.

Status
Implemented. Current release/branch: spring_dev
Final software architecture for SPRING-ARI

This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Source code repository:
https://gitlab.inria.fr/spring/wp5_spoken_conversations/interaction.git
SUBFOLDER: interaction_manager

Inputs
Input: TF

Topic subscription: /nlp_node/nlu [JSON String/JSON String]

Input: input

Topic subscription: /register_server [service/service]

Topic subscription: /interaction_manager/update [service/service]

Topic subscription: /interaction_manager/query [service/service]

Input: /dialogue_start/feedback

Topic subscription: /controller_status [ControllerStatus/ControllerStatus]

Input: semantic scene description [ON HOLD]

Input: /dialogue_start/status

Topic subscription: /humans/persons/<id>/softbiometrics [hri_msgs/Softbiometrics]

Outputs
Topic publication: /task*_ros_server_action [action/action]

Output: /dialogue_start

Output: /navigation goals

Output: /look_at goals

Output: gestures

Topic publication: /RPN [action/action]

Output: /social_strategy_supervisor_server/goal

Dependencies
action/action
std_msgs/Empty

JSON String/JSON String
ControllerStatus/ControllerStatus

hri_msgs/Softbiometrics

**nlp_node**

The node *nlp_node* (id: nlp_node) is maintained by HWU.

Status
Implemented. Current release/branch: 0.0.1
Source code repository:

Inputs
Topic subscription: /nlp_node/get_answer [service/service]

Outputs
Topic publication: /nlp_node/nlu [JSON String/JSON String]

Dependencies
service/service

JSON String/JSON String

**recipe_planner**

The node *recipe_planner* (id: recipe_planner) is maintained by HWU.

Status
Implemented. Current release/branch: spring_dev
Source code repository: https://gitlab.inria.fr/spring/wp5_spoken_conversations/plan_actions.git

Inputs
Input: semantic scene description
Topic subscription: /task*_ros_server_action [action/action]

Input: PDDL yaml library

Topic subscription: /humans/persons/<id>/softbiometrics [hri_msgs/Softbiometrics]

Outputs
Output: /queries
Output: /updates

Topic publication: /RPN [action/action]
This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Dependencies
std_msgs/Empty
action/action
service/service
hri_msgs/Softbiometrics

ros_petri_net_node

The node ros_petri_net_node (id: ros_petri_net_node) is maintained by HWU.
Status
Implemented. Current release/branch: 0.0.1
Source code repository:
https://gitlab.inria.fr/spring/wp5_spoken_conversations/ros_petri_net_planner
Inputs
Topic subscription: /RPN [action_server/action_server]

Outputs
Dependencies
action_server/action_server

social_state_analyzer

The node social_state_analyzer (id: social_state_analyzer) is maintained by HWU.
Status
Implemented. Current release/branch: 0.0.1
Source code repository:
Inputs
Input: /h/p/tracked
Topic subscription: /kb/query [service/service]
Input: /h/p/*
Input: /h/b/tracked
Input: /h/b/*
Input: /h/f/*
Topic subscription: /social_state_analyzer_server [action_server/action_server]
Input: /h/f/tracked

Outputs
Topic publication: /social_state_analyser/state [std_msgs/String]

Dependencies
std_msgs/Empty

service/service

std_msgs/String

action_server/action_server

social_strategy_supervisor

The node social_strategy_supervisor (id: social_strategy_supervisor) is maintained by HWU.

Status
Implemented. Current release/branch: 0.0.1

Source code repository:

Inputs
Topic subscription: /social_state_analyser/state [std_msgs/String]

Topic subscription: /social_strategy_supervisor_server [action_server/action_server]

Outputs
Output: /look_at goals

Output: /go_to goals

Dependencies
std_msgs/Empty

std_msgs/String

action_server/action_server

INRIA
basestation_republisher

The node basestation_republisher (id: basestation_republisher) is maintained by INRIA.

Status
Implemented. Current release/branch: master

Source code repository: https://gitlab.inria.fr/spring/wp3_av_perception/docker_republish

Inputs
Input: /torso_front_camera/aligned_depth_to_color/image_raw/theora

Topic subscription: /front_camera/fisheye/image_raw/compressed [sensor_msgs/CompressedImage]

Input: /torso_front_camera/color/image_raw/theora

Input: /head_front_camera/color/image_raw/compressed

Outputs
Topic publication: /*_basestation/head_front_camera/... [sensor_msgs/Image]

Topic publication: /*_basestation/fisheye/... [sensor_msgs/Image]

Dependencies
sensor_msgs/Image
std_msgs/Empty
sensor_msgs/CompressedImage

**body_3d_tracker**

*The node body_3d_tracker (id: body_3d_tracker) is maintained by INRIA.*

Status
Implemented. Current release/branch: main

Source code repository:

Inputs
Topic subscription: /humans/bodies/<id>/skeleton2d [hri_msg/Skeleton2D]

Topic subscription: /front_camera/fisheye/image_raw [sensor_msgs/Image]

Topic subscription: /tracker/tracker_output [std_msgs/String]

Outputs
Output: tf: /body_* (tf)

Topic publication: /humans/bodies/tracked [hri_msgs/IdsList]

Dependencies
tf/transform_broadcaster

hri_msg/Skeleton2D

sensor_msgs/Image
The node `body_to_face_mapper` (id: `body_to_face_mapper`) is maintained by INRIA.

Status
Implemented. Current release/branch: main

Source code repository: node associates detected bodies to detected faces in image-space.
REPO: https://gitlab.inria.fr/spring/wp3_av_perception/body_to_face_mapper

Inputs
Topic subscription: `/humans/faces/tracked` [hri_msgs/IdsList]
Topic subscription: `/humans/bodies/tracked` [hri_msgs/IdsList]
Topic subscription: `/humans/faces/TEST_ID_FACE/roi` [hri_msgs/NormalizedRegionOfInterest2D]
Topic subscription: `/humans/bodies/<id>/roi` [hri_msgs/NormalizedRegionOfInterest2D]

Outputs
Topic publication: `/humans/candidate_matches` [hri_msgs/IdsMatch]

Dependencies
hri_msgs/IdsList
hri_msgs/IdsMatch
hri_msgs/NormalizedRegionOfInterest2D

The node `front_fisheye_2d_body_pose_detector_op` (id: `front_fisheye_2d_body_pose_detector_op`) is maintained by INRIA.

Status
Implemented. Current release/branch: main

Source code repository: https://gitlab.inria.fr/spring/wp3_av_perception/front_fisheye_2d_body_pose_detector_op

Inputs
Topic subscription: `/tracker/tracker_output` [std_msgs/String]
Topic subscription: `/front_camera/fisheye/image_raw` [sensor_msgs/Image]
This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

**Outputs**

Topic publication: /humans/bodies/<id>/skeleton2d [hri_msg/Skeleton2D]

**Dependencies**

hri_msg/Skeleton2D

std_msgs/String

sensor_msgs/Image

---

**front_fisheye_2d_body_tracker**

*The node *front_fisheye_2d_body_tracker* (id: front_fisheye_2d_body_tracker) *is maintained by INRIA.*

**Status**

Implemented. Current release/branch: devel

**Source code repository:** [https://gitlab.inria.fr/spring/wp3_av_perception/multi-person_visual_tracker/](https://gitlab.inria.fr/spring/wp3_av_perception/multi-person_visual_tracker/)

**Inputs**

Topic subscription: /front_camera_basetation/fisheye/image_raw/compressed [sensor_msgs/CompressedImage]

**Outputs**

Output:

Topic publication: /humans/bodies/<id>/cropped [sensor_msgs/Image]

Topic publication: /tracker/tracker_output [std_msgs/String]

Topic publication: /humans/bodies/<id>/roi [hri_msgs/NormalizedRegionOfInterest2D]

**Dependencies**

std_msgs/Empty

sensor_msgs/Image

sensor_msgs/CompressedImage

std_msgs/String

hri_msgs/NormalizedRegionOfInterest2D

---

**go_to_body_action_server**

*The node *go_to_body_action_server* (id: go_to_body_action_server) *is maintained by INRIA.*
This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

### Implemented. Current release/branch: 0.0.1
Source code repository:

**Inputs**
Input: goal

**Topic subscription**: /controller_status  [ControllerStatus/ControllerStatus]

**Outputs**
Topic publication:  /go_towards  [GoTowards/GoTowards]

**Dependencies**
std_msgs/Empty
GoTowards/GoTowards
ControllerStatus/ControllerStatus

---

**go_to_group_action_server**

*The node go_to_group_action_server (id: go_to_group_action_server) is maintained by INRIA.*

**Status**
Implemented. Current release/branch: 0.0.1
Source code repository:

**Inputs**
Input: goal

**Topic subscription**: /controller_status  [ControllerStatus/ControllerStatus]

**Outputs**
Topic publication:  /go_towards  [GoTowards/GoTowards]

**Dependencies**
GoTowards/GoTowards
std_msgs/Empty
ControllerStatus/ControllerStatus

---

**go_to_person_action_server**

*The node go_to_person_action_server (id: go_to_person_action_server) is maintained by INRIA.*

**Status**
Implemented. Current release/branch: 0.0.1
Source code repository:
Inputs
Input: goal

Topic subscription: /controller_status [ControllerStatus/ControllerStatus]

Outputs
Topic publication: /go_towards [GoTowards/GoTowards]

Dependencies
GoTowards/GoTowards
std_msgs/Empty
ControllerStatus/ControllerStatus

go_to_position_action_server

The node go_to_position_action_server (id: go_to_position_action_server) is maintained by INRIA.
Status
Implemented. Current release/branch: 0.0.1
Source code repository:
Inputs
Input: goal

Topic subscription: /controller_status [ControllerStatus/ControllerStatus]

Outputs
Topic publication: /go_towards [GoTowards/GoTowards]

Dependencies
std_msgs/Empty
GoTowards/GoTowards
ControllerStatus/ControllerStatus

group_detector

The node group_detector (id: group_detector) is maintained by INRIA.
Status
Implemented. Current release/branch: main
Source code repository: https://gitlab.inria.fr/spring/wp4_behavior/group_detector
Inputs
Input: /h/p/tracked

Input: tf: /person_* (tf)

Outputs
Topic publication: /h/g/tracked [hri_msgs/IdList]

Output: tf: /group_* (tf)

Topic publication: /humans/group/<id>/ [hri_msgs/IdList]

Dependencies
hri_msgs/IdList
tf/transform_broadcaster
std_msgs/Empty
tf/transform_listener

**look_at_action_server**

*The node look_at_action_server (id: look_at_action_server) is maintained by INRIA.*

Status
Implemented. Current release/branch: 0.0.1

Source code repository:
Inputs
Input: goal

Topic subscription: /controller_status [ControllerStatus/ControllerStatus]

Outputs
Topic publication: /look_at [LookAt/LookAt]

Dependencies
std_msgs/Empty

LookAt/LookAt
ControllerStatus/ControllerStatus

**look_at_person_server**

*The node look_at_person_server (id: look_at_person_server) is maintained by INRIA.*

Status
Implemented. Current release/branch: 0.0.1
## Look at Position Action Server

*The node look_at_position_action_server (id: look_at_position_action_server) is maintained by INRIA.*

**Status**
- Implemented. Current release/branch: 0.0.1

**Source code repository:**
- Inputs
  - Input: goal

<table>
<thead>
<tr>
<th>Topic Subscription</th>
<th>Topic Publication</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>/controller_status</td>
<td>/navigate</td>
<td>std_msgs/Empty</td>
</tr>
<tr>
<td>[ControllerStatus/ControllerStatus]</td>
<td>[Navigate/Navigate]</td>
<td>LookAt/LookAt, ControllerStatus/ControllerStatus</td>
</tr>
</tbody>
</table>

## Occupancy Map Republisher

*The node occupancy_map_republisher (id: occupancy_map_republisher) is maintained by INRIA.*

**Status**
- This node is not yet implemented.

**Inputs**
- The node occupancy_map_republisher (id: occupancy_map_republisher) is maintained by INRIA.

<table>
<thead>
<tr>
<th>Topic Subscription</th>
<th>Topic Publication</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>/controller_status</td>
<td></td>
<td>std_msgs/Empty</td>
</tr>
<tr>
<td>[ControllerStatus/ControllerStatus]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Navigate Node

**Status**
- Implemented. Current release/branch: 0.0.1

**Source code repository:**
- Inputs
  - Input: goal

<table>
<thead>
<tr>
<th>Topic Subscription</th>
<th>Topic Publication</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>/controller_status</td>
<td>/navigate</td>
<td>std_msgs/Empty</td>
</tr>
<tr>
<td>[ControllerStatus/ControllerStatus]</td>
<td>[Navigate/Navigate]</td>
<td>LookAt/LookAt, ControllerStatus/ControllerStatus</td>
</tr>
</tbody>
</table>

---

This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.
This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Topic subscription: /slam/obstacle_map [OccupancyGrid/OccupancyGrid]

Topic subscription: /slam/occupancy_map [OccupancyGrid/OccupancyGrid]

Outputs
Topic publication: /slam/local_map [OccupancyGrid/OccupancyGrid]

Dependencies
OccupancyGrid/OccupancyGrid

pointcloud2occmap

The node pointcloud2occmap (id: pointcloud2occmap) is maintained by INRIA.

Status
This node is not yet implemented.

Inputs
Input: torso pointcloud

Outputs
Topic publication: /slam/obstacle_map [OccupancyGrid/OccupancyGrid]

Dependencies
std_msgs/Empty

OccupancyGrid/OccupancyGrid

social_mpc

The code is primarily developed at INRIA by Timothée Wintz.

The node social_mpc (id: social_mpc) is maintained by INRIA.

Status
Implemented. Current release/branch: devel

Source code repository: https://gitlab.inria.fr/spring/wp6_robot_behavior/robot_behavior

Inputs
Input: /h/p/tracked

Topic subscription: /look_at [LookAt/LookAt]

Topic subscription: /go_towards [GoTowards/GoTowards]

Input: tf: /body_* (tf)

Input: /h/b/tracked

Input: tf: /person_* (tf)
This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Topic subscription: /navigate [Navigate/Navigate]

Topic subscription: /slam/local_map [OccupancyGrid/OccupancyGrid]

Input: /h/g/tracked

Input: /joint_states

Input: tf: /group_* (tf)

Outputs
Output: /controller_status

Output: tf: /final_point /nav_goal... (tf)

Topic publication: /nav_vel [Twist/Twist]

Topic publication: /head_controller/command [JointTrajectory/JointTrajectory]

Dependencies
std_msgs/Empty

LookAt/LookAt

GoTowards/GoTowards

tf/transform_listener

Navigate/Navigate

OccupancyGrid/OccupancyGrid

tf/transform_broadcaster

Twist/Twist

JointTrajectory/JointTrajectory

voicebodycandidatematches

The node voicebodycandidatematches (id: voicebodycandidatematches) is maintained by INRIA.

Status
Implemented. Current release/branch: 0.0.1

Source code repository:
Input: /humans/voices/tracked

Input: tf: /voice_* (tf)

Input: /humans/bodies/tracked

Input: tf: /body_* (tf)

Outputs
Topic publication: /humans/candidate_matches [hri_msgs/IdsMatch]

Dependencies
std_msgs/Empty
tf/transform_listener
hri_msgs/IdsMatch

Other

PAL
fisheye

The node fisheye (id: fisheye) is maintained by PAL.
Status
Implemented. Current release/branch: 0.0.1
Source code repository:

Inputs

Outputs
Topic publication: /torso_front_camera/color/image_raw [sensor_msgs/Image]

Dependencies
sensor_msgs/Image

hri_person_manager

The node hri_person_manager (id: hri_person_manager) is maintained by PAL.
Status
Implemented. Current release/branch: master
Source code repository: https://gitlab.inria.fr/spring/wp7_ari/hri_person_manager.git

Inputs

Topic subscription: /humans/candidate_matches [hri_msgs/IdsMatch]

Outputs
Output: /h/p/...

Topic publication: /h/p/tracked [hri_msgs/IdsList]

Output: tf /person_* (tf)

Dependencies
std_msgs/Empty
hri_msgs/IdsList
tf/transform_broadcaster
hri_msgs/IdsMatch

知识核心

The node knowledge_core (id: knowledge_core) is maintained by PAL.

Status
Implemented. Current release/branch: 2.8.0
Source code repository: https://gitlab.inria.fr/spring/wp7_ari/knowledge_core

Inputs
Topic subscription: /kb/add_fact [std_msgs/String]

Outputs
Output: /kb/query [service]

Dependencies
std_msgs/String
std_msgs/Empty

人物事实

The node people_facts (id: people_facts) is maintained by PAL.

Status
Implemented. Current release/branch: 0.2.2
Source code repository: https://gitlab.inria.fr/spring/wp7_ari/people_facts

Inputs
Input: /h/p/...

Outputs
Topic publication: /kb/add_fact [std_msgs/String]

Dependencies
The node raspicam (id: raspicam) is maintained by PAL.
Status
Implemented. Current release/branch: 0.0.1
Source code repository:
Inputs
Outputs
Topic publication: /head_front_camera/color/image_raw [sensor_msgs/Image]
Dependencies
sensor_msgs/Image

The node respeaker_ros (id: respeaker_ros) is maintained by PAL.
Status
Implemented. Current release/branch: master
Source code repository:
Inputs
Outputs
Topic publication: /audio/raw_audio [audio_common_msgs/AudioData]
Dependencies
audio_common_msgs/AudioData

The node Robot functional layer (id: robotfunctionallayer) is maintained by PAL.
Status
Implemented. Current release/branch: 0.0.1
Source code repository:
Inputs
Input: input

Topic subscription: /tts [action_server/action_server]
Outputs
Output: /joint_states

Dependencies
std_msgs/Empty

action_server/action_server
torso_rgbd_camera
The node torso_rgbd_camera (id: torso_rgbd_camera) is maintained by PAL.
Status
Implemented. Current release/branch: 0.0.1
Source code repository:
Inputs
Outputs
Output: pointcloud
Output: torso_front_camera/infra/
Output: /torso_front_camera/imu

Topic publication: /torso_front_camera/color/image_raw [sensor_msgs/Image]

Dependencies
std_msgs/Empty

sensor_msgs/Image

UNITN
activity_recognition
The node activity_recognition (id: activity_recognition) is maintained by UNITN.
Status
Implemented. Current release/branch: 0.0.1
Source code repository:
Inputs
Topic subscription: /vision_msgs/human_2d_pose
[human_2d_pose_estimation/Frame]
Outputs
Output: [?] output to be defined + added to hri_msgs if possible

Dependencies
std_msgs/Empty

human_2d_pose_estimation/Frame

depth_estimation
The node depth_estimation (id: depth_estimation) is maintained by UNITN.
This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

**Status**
Implemented. Current release/branch: main

**Source code repository:** [https://gitlab.inria.fr/spring/wp4_behavior/depth-estimation](https://gitlab.inria.fr/spring/wp4_behavior/depth-estimation)

**Inputs**
Topic subscription: /*_basestation/head_front_camera/... [sensor_msgs/Image]

**Outputs**
Topic publication: /depth_estimation [sensor_msgs/Image]

**Dependencies**
sensor_msgs/Image

---

**emotion_estimation**

*The node emotion_estimation (id: emotion_estimation) is maintained by UNITN.*

**Status**
Implemented. Current release/branch: 0.0.1

**Source code repository:**

**Inputs**
Topic subscription: /humans/faces/TEST_ID_FACE/cropped [sensor_msg/Image]

**Outputs**
Topic publication: /humans/faces/TEST_ID_FACE/expression [hri_msgs/Expression]

**Dependencies**
sensor_msg/Image

hri_msgs/Expression

---

**face_tracker**

*The node face_tracker (id: face_tracker) is maintained by UNITN.*

**Status**
Implemented. Current release/branch: 0.0.1

**Source code repository:** [https://gitlab.inria.fr/spring/wp4_behavior/face-tracker](https://gitlab.inria.fr/spring/wp4_behavior/face-tracker)

**Inputs**
Topic subscription: /*_basestation/head_front_camera/... [sensor_msgs/Image]

**Outputs**
Topic publication: /humans/faces/TEST_ID_FACE/cropped [sensor_msg/Image]

Topic publication: /humans/faces/TEST_ID_FACE/roi [hri_msgs/NormalizedRegionOfInterest2D]

Topic publication: /humans/faces/tracked [hri_msgs/IdsList]
Dependencies

sensor_msg/Image

sensor_msgs/Image

hri_msgs/NormalizedRegionOfInterest2D

hri_msgs/IdsList

gaze_estimation

*The node gaze_estimation (id: gaze_estimation) is maintained by UNITN.*

Status
Implemented. Current release/branch: devel

Source code repository:

Inputs

Topic subscription: /*_basestation/head_front_camera/... [sensor_msgs/Image]

Topic subscription: /depth_estimation [sensor_msgs/Image]

Topic subscription: /humans/faces/TEST_ID_FACE/roi [sensor_msgs/RegionOfInterest]

Outputs

Output: GazeFrame [2D point in rgb frame]

Dependencies

sensor_msgs/Image

std_msgs/Empty

sensor_msgs/RegionOfInterest

human_2d_pose_estimation

*The node human_2d_pose_estimation (id: human_2d_pose_estimation) is maintained by UNITN.*

Status
Implemented. Current release/branch: main

Source code repository: https://gitlab.inria.fr/spring/wp4_behavior/human-2d-pose-estimation

Inputs

Topic subscription: /*_basestation/head_front_camera/... [sensor_msgs/Image]

Outputs
Topic publication: /vision_msgs/human_2d_pose
  [human_2d_pose_estimation/Frame]

Dependencies
sensor_msgs/Image
human_2d_pose_estimation/Frame

mask_detector

The node mask_detector (id: mask_detector) is maintained by UNITN.
Status
Implemented. Current release/branch: master
Source code repository: https://gitlab.inria.fr/spring/wp4_behavior/mask-detection
BIN:mask_detector.py
Inputs
Topic subscription: /humans/faces/TEST_ID_FACE/cropped    [sensor_msg/Image]

Outputs
Topic publication: /humans/faces/TEST_ID_FACE/has_mask    [wp4_msgs/FaceMask]

Dependencies
sensor_msg/Image
wp4_msgs/FaceMask

soft_biometrics_estimator

The node soft_biometrics_estimator (id: soft_biometrics_estimator) is maintained by UNITN.
Status
Implemented. Current release/branch: master
Source code repository: https://gitlab.inria.fr/spring/wp4_behavior/wp4_behavior_understanding
SUBFOLDER:wp4_people_characteristics BIN:soft_biometrics_estimator.py
Inputs
Topic subscription: /*_basestation/head_front_camera/...    [sensor_msgs/Image]

Topic subscription: /humans/faces/tracked    [hri_msgs/IdsList]

Topic subscription: /humans/faces/TEST_ID_FACE/roi
  [hri_msgs/NormalizedRegionOfInterest2D]

Outputs
Topic publication: /humans/candidate_matches [hri_msgs/IdsMatch] [face reco/face reco]

Topic publication: /humans/faces/TEST_ID_FACE/softbiometrics [hri_msgs/SoftBiometrics]

Dependencies
face reco/face reco

sensor_msgs/Image

hri_msgs/IdsList

hri_msgs/NormalizedRegionOfInterest2D

hri_msgs/SoftBiometrics