Grant agreement no: FP7-600877

SPENCER:

Social situation-aware perception and action for cognitive robots

Project start: April 1, 2013

Duration: 3 years

DELIVERABLE 6.4

Final demonstration at Amsterdam Schiphol Airport

Due date: month 35 (February 2016)

Lead contractor organization: ALU-FR

Dissemination Level: PUBLIC

Contents

1 Introduction **3**

2 Integration Week IVb **3**

3 Integration Week V and Final Deployment **4**

3.1 Group Guidance 6
3.2 Socially-Aware Motion Planner 6
3.2.1 A little rude is more efficient 6
3.2.2 Multi-hypothesis path planner 6
3.2.3 Socially-aware elastic band 8
3.3 Interaction Unit 8
3.3.1 Graphical user interface 8
3.3.2 State machines for task execution 9
3.3.3 Remote operator interface 10
3.3.4 Boarding Pass Reader Integration 10
3.4 Mapping and Localization 13
3.5 People Analysis 13

4 Visual Impressions **14**

Abstract

This deliverable reports on the final demonstration of the SPENCER robot at Schiphol Airport which is the outcome of the fifth and final integration week and the final Milestone MS4 "Deployment at Amsterdam Schiphol Airport". The goal is the mature integration of all modules and that the overall system fulfills the specification of the use-case as defined in collaboration with end-user KLM in WP1. All components should be in their final functional state and satisfy the success measures defined in their respective work package.

1 Introduction

The overall goal of the final SPENCER integration week and deployment was to finalize, improve, and extensively test the integration of the components of the architecture. The outcome of this integration week was defined to be project milestone MS4, for which the DoW specifies the following:

> *Final demonstration of the SPENCER platform at the Amsterdam Schiphol Airport. The robot fulfills the entire specification of the use-case as defined in collaboration with KLM in WP1. Final integration of all modules. This is the output of integration week V. All components are in their final functional state and satisfy the success measures defined in their respective work package*

In addition to plenary integration weeks VI and V, several unplanned actions had been taken prior to the final deployment. They include:

- Pre-integration week for safety and perception (Aug 10 to 14, 2015): Partners ALU-FR, ORU and CNRS met in Toulouse to solve the lower-level problems regarding safety and to integrate and robustify robot environment and human perception components.
- Pre-integration week for planning and supervision (Aug 17 to 21, 2015): Partners ALU-FR and CNRS met in Toulouse to integrate the motion planning and supervision components.
- Transport of the robot platform to partner site ALU-FR (instead of CNRS) after integration week IV. Main reason: larger and more realistic test environment.
- Plenary integration week IVb (Nov 30 to Dec 5, 2015): first Schiphol deployment and tests.
- Pre-integration phase in Freiburg (two months, Jan to Feb 2015): Partners CNRS, UT, ORU visit ALU-FR. Complementary and intense tests of the robot basic and advanced functions in a larger and populated environment (foyer of a lecture hall). User study carried out by UT and ALU-FR.

2 Integration Week IVb

Plenary integration week IVb was initially unplanned and organized after a Steering Board decision during Steering Board meeting V in fall 2015. It took place in Schiphol from November 30 to December 5, 2015. This was the first deployment of the SPENCER robot in the Schiphol Airport and one of its main goals was gaining experience in the environment as a risk mitigation measure for the final deployment.

Summarizing, the main activities and achievements during integration week IVb are as follows:

- The SPENCER system was deployed in a real-world crowded environment for the first time. All core components of the architecture required for the targeted functionalities including supervision system, mapping and localization components, global and local motion planners, people detection and tracking system were fully integrated and tested.
- The first two days of the integration week were mainly used to collect large datasets for person tracking and analysis, social navigation as well as for on-line object classification.
- Initial maps of the Pier-B and C were generated after collecting 3D laser data during the night shifts. These maps were subsequently used for navigation.
- The boarding pass reader was integrated and tested with passes generated by KLM's IT department.
- Also many unforeseen issues have been observed during this integration week, e.g. sparkly floors that disturb the RGB-D sensors or an insufficient number of free MAC addresses for the network to work in a large integration team. Therefore, integration week IVb turned out to be very helpful and gave sufficient time to the consortium to address these issues before the final integration.

3 Integration Week V and Final Deployment

The final integration week V took place between March 10 to 24, 2016. The main purpose were final tests, deployment and demonstration.

All basic and many advanced components were successfully tested on to the SPENCER robot with crowded environment to give a fully functional guide robot with advanced perception, decision, learning and control capabilities. Pier-B, C and the shopping area after the Schengen passport control of the Schiphol airport was accurately mapped using 3D laser data that was consequently used for navigation during the final demonstrations. The map was annotated with semantic information such as gates, toilets and other places-of-interest information. Towards the end of the integration week the robot reached excellent performance in terms of smooth, high speed yet safe navigation in crowded environments. Together with an appealing user interface and the guiding supervision system the robot was used to guide multiple passengers from information desk area to gates in Pier-B of the Schiphol airport. We conducted user studies showing high acceptance and positive response towards the design and the behavior of the robot among real passengers at the airport.

We also collected more data sets that will help consortium partners to further improve perception and control algorithms. Please navigate to `https://www.youtube.com/watch?v=ir_Ku4rCOO8` for a YouTube Video that demonstrates results and achievements of SPENCER robot accomplished during the final integration week (see also Fig. 7 for screenshots).

Fig. 1 shows the integration level of all components in the SPENCER architecture. The green modules are the components that have been fully integrated, tested and demonstrated during the final integration week.

We will now mention particular aspects, lessons learned or adaptations that we have experienced or made during the final integration week in individual components or subsystems:

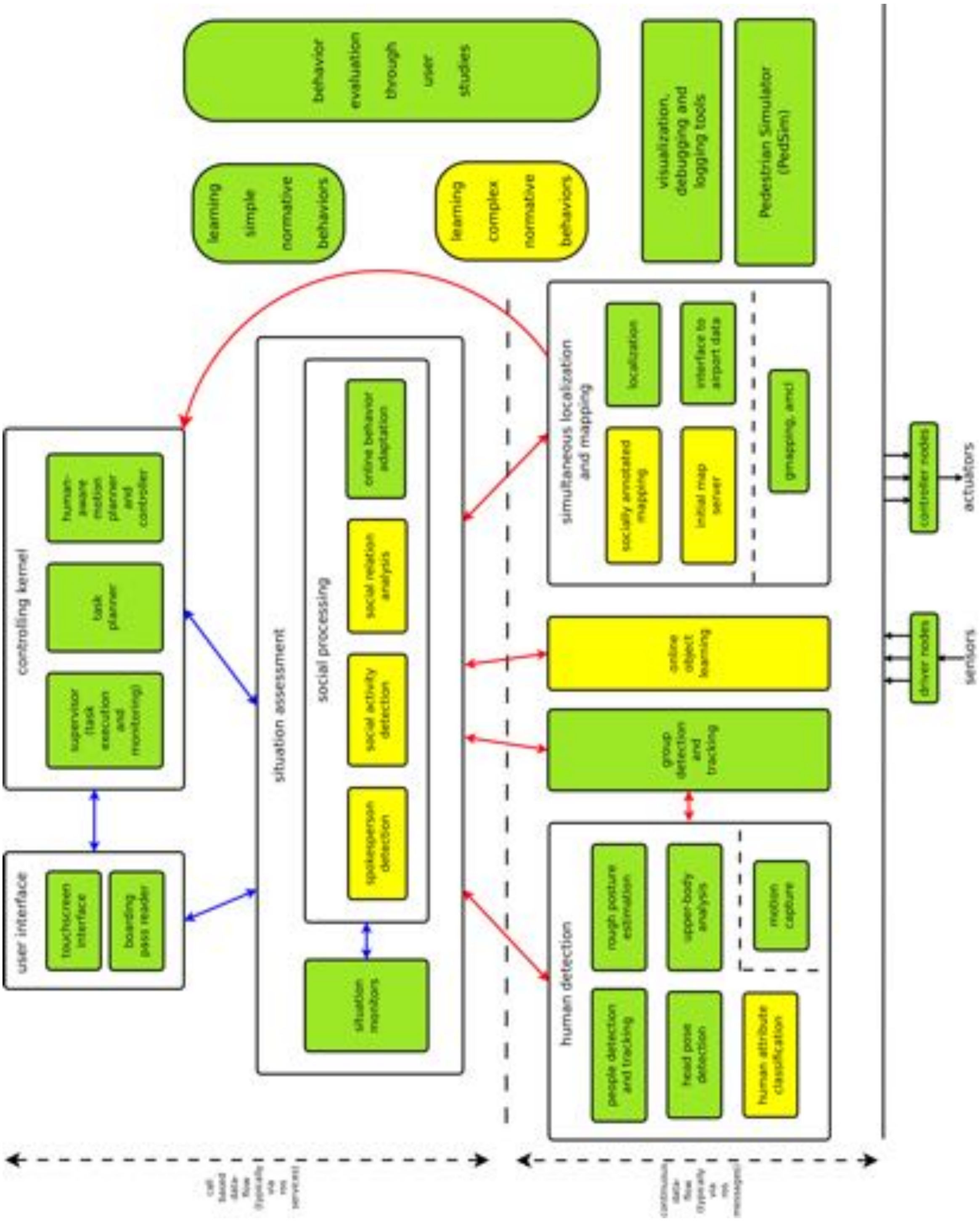

Figure 1: Final SPENCER architecture. The green modules are the components that have been fully integrated and tested during the final integration week. The yellow components have not been fully integrated, however, research in the respective tasks has been carried out successfully leading to several SPENCER publications.

3.1 Group Guidance

Guiding a group in a human-aware way is a complex task in an environment such as Schiphol airport. A significant aspect that we noticed during the tests in the airport was that some people often did not really “follow” the robot. We can assume that most passengers have had very little prior experience with robots. We noticed that a large group of users – typically transfer passengers with generous connecting times – that were supposed to be guided by the robot would instead move in front of it to take pictures or try to hug it. They seemed more interested in the robot itself than in reaching their destination. Such passengers were then wongly classified as “not following the robot” which should probably be handled by a more “relaxed” notion of “engaged users”.

The experience also showed that users have different preferences in how to follow the robot. Some people prefer following the platform, others prefer staying side-by-side with the robot. Accounting for such preferences could also be incorporated into the current algorithms with little extra effort.

3.2 Socially-Aware Motion Planner

The socially-aware motion planner, deployed during the final demonstration, was improved by the experiences made during Integration Week IVb in the following ways:

3.2.1 A little rude is more efficient

We found that an airport is not only a very crowded environment, it is also a place where humans are very goal-oriented, partly in a rush, and less attentive of what is happening around them. For those passengers, even an intelligent robot that usually attracts a lot of attention, becomes a litte-noticed object to which they do not pay particular attention. Our tests during the final integration week led to the conclusion that under such conditions a robot that fully respects all learned social norms (e.g. in the learned “polite behavior”), ends up in being overly reactive and partly freeze in highly dynamic crowds of fast moving people. To reduce the number of such freeze-events, we activated the learned “rude behavior” in which the robot ignores many social norms which significantly improved the efficiency of the robot and reduced the number of missions aborted for planning problems to zero. By this, we have confirmed the human behavior *the more we are in a rush the more social rules we are ignoring*.

3.2.2 Multi-hypothesis path planner

One of the limits of the pre-Integration Week IVb motion planner were direction oscillations while the robot was driving through a crowd. Such oscillations caused detours of the robot e.g. when it was trying to avoid a crowd as one big obstacle. The rude behavior (see above subsection) helped to reduce this robot behavior by forcing the platform to be more efficient than socially aware. The most effective counteraction was, however, a novel multi-hypothesis path planner developed in SPENCER. This planner considers the environment in two different conditions: first, modeled only by static obstacles (the ones stored in the grid map generated off-line) and second modeled by the static obstacles and all humans/dynamic obstacles. For both conditions, a path is planned and then compared. By exploiting differences in the two paths (path length, discrete Fréchet distance, homotopy class test)

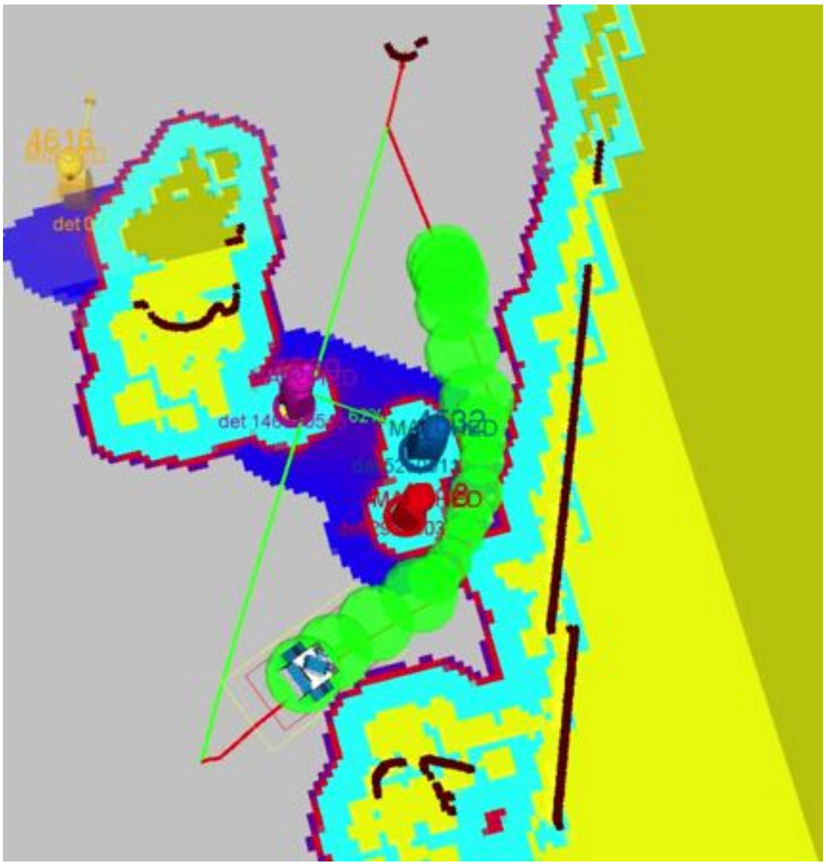

Figure 2: Socially aware motion planning. This figures shows two possible ways to pass a crowd (here: three persons where one takes a foto of the other two). The blue area on the floor shows the social relationship between the group members, the red line is the social-aware path computed and selected by our system, and the green line is the static (direct) path. Green spheres represent the elastic band.

and perception information (pose of people tracks), we evaluate the two hypotheses and choose the best path accordingly (see Fig.2):

- There is a large crowd in front of the robot, and instead of avoiding it the robot keeps going on the path generated considering only the static obstacles and activating an HRI action (*static* path).
- The space in front of the robot is not fully obstructed by the crowd and the robot can follow the new path without encountering large detours or oscillations (*social-aware* path)

Each of the two paths (*static* path and *social-aware* path) is generated using the Dijkstra algorithm and considering a different cost-map for each case. Once the global path planner has chosen between the two hypotheses, it will send the path to follow to the local planner together with the type of the taken choice, so that the Elastic Band (the current local planner) could select the proper cost-map to plan on and it will activate the proper HRI action. This planner runs with a frequency of 0.1 Hz and it is asked to replan when the elastic band cannot generate velocity commands.

3.2.3 Socially-aware elastic band

The multi-hypothesis path planner is coupled with an Elastic Band (eband) local path planner. The eband planner generates velocity commands such that the robot does not collide with obstacles and humans, the robot actions are legible by the humans, the robot follows the path generated by the global path planner and activates the HRI action accordingly, the commands do not violate the dynamic constraints of the actuators.

The elastic band generally is meant as a planner that adapts the plan based on the obstacles position and in general on the changes in the environment. Each obstacle pushes away the band by means of an elastic force. We modified an existing open-source ROS package version of the elastic band to better couple it with our multi-hypothesis global path planner and to generate legible motions.

Each time that the global path planner provides a new path, the eband planner selects the proper local cost-map based on the hypothesis that the robot is following. In case that the *static* path is selected the eband activates its HRI module: every time that d (which is a parameter) people tracks are in front of the robot direction (and stopping the robot), the HRI action is triggered to avoid that the robot is kept there frozen. During the final demonstration the HRI action consisted on reproducing a simple string: the robot was asking for permission to go through, by saying "Excuse me".

To improve legibility of the robot motions, the velocity of the robot, generated by the eband planner is scaled based on the distance of the laser scans readings to the current band (see Fig.2). For this task, to reduce the oscillations in terms of translational velocity, we did not adopt people tracking information as it was in the previous *legibility motions* component tested during Integration week IV. To improve legibility a priori we reduced the accelerations for translational and angular velocity too.

This planner runs with a frequency of 6 Hz.

3.3 Interaction Unit

A subsystem for user interaction was developed by ALU-FR during the final year of the project. This subsystem, comprised of four major modules described in the following, was originally not promised as part of the DoW. However, it was necessary for the final demonstration at the airport and the preceding tests in integration week IVb and in the foyer of the lecture hall at ALU-FR.

3.3.1 Graphical user interface

The GUI is based upon PyQt and `rospy` and consists of a total of 54 individual screens (pages). It has been designed with a clear layout (high contrast, large buttons and fonts) to allow readability even when the user is a few meters away from the robot, e. g. while following during group guidance, and under direct sunlight. For the final demonstration, only English language support was added, but it would easily be possible to add support for further languages in the future.

Six different operation modes (use cases) have been implemented in the GUI:

- **Schengen-to-Gate guidance:** The main mode used during the final demonstration. Initially, the robot is stationary, waits for passengers to touch the screen, lets them scan their boarding passes, and then starts guiding them to their departure gate once they press a button.

b 

Figure 3: Two different screens of the graphical user interface shown on the touchscreen display. The GUI is based upon rospy and PyQt.

- **Gate-to-Gate guidance:** The robot drives to a gate, notifies waiting passengers that there has been a gate change, and then guides all passengers who follow to their new departure gate. While this use case was fully implemented in the GUI and state machine (see Sec. 3.3.2), there was insufficient time to test it at Schiphol and it was thus not part of the final demonstration and evaluation.
- **Information kiosk:** Here, the user can either ask the robot to show information about the SPENCER project on the screen (research goals, some video showcases etc.), or either manually select one out of four hardcoded points of interests to be guided to. This mode was mainly used for testing navigation and group guidance when the boarding pass reader integration was not yet completed.
- **Move to location:** Drives the robot to an annotated point of interest, specified via the remote operator interface (Sec. 3.3.3) with the screen facing in driving direction (i.e. not guiding a group).
- **Out of service:** Just shows an "Out of service" message on the touchscreen. This proved to be very helpful at the airport, to prevent passengers from trying to use the robot while it was undergoing maintenance or debugging.
- **Recharge battery:** Activated automatically when the battery level reaches close to critical level. Shows a warning message on the screen and cancels any other previously active operation mode. In the future, in this mode, the robot could automatically drive to a possible recharging station.

Two different screens of the graphical user interface are shown in Fig. 3.

3.3.2 State machines for task execution

Each use case essentially represents a separate workflow, or process. The process logic is, for better separation of concerns, not implemented within the GUI itself, but inside a separate set of high-level state machines, using SMACH and smach_ros. This existing framework already includes visualization capabilities, that are highly useful to debug the complex state machines and the integration with the GUI and other system components.

The SMACH-based state machines represent the glue code that integrates the different modules of the interaction subsystem, discussed in this section, with the supervisor functions (group guidance, move-to-goal etc.).

Fig. 4 (top) shows the top-level state machine, containing all 6 operation modes (use cases) described above. As can be seen, even a single operation mode, such as the Schengen-to-Gate use case shown in the middle of Fig. 4, is fairly complex and contains a large number of transitions. For this reason, using SMACH was very helpful in this case as it allows reuse of different lower-level state machines (such as the group guidance process, shown in Fig. 4 on the bottom) across various higher-level state machines (i. e. operation modes). Still, rigorous testing was required (by essentially testing all possible transitions, fixing issues that arose, and re-iterating) during the final integration week to ensure a solid user experience.

3.3.3 Remote operator interface

During the pre-integration phase at ALU-FR, a web browser-based interface for robot remote operation via wi-fi was developed (see Fig. 5). This interface is based upon ros_bridge (using HTML and JavaScript to interface with a webserver running on the robot), and serves multiple purposes: 1) giving a quick diagnostic overview of the robot, to see why e. g. the robot is currently not able to move, 2) viewing the robot's camera streams, to see if all sensors are properly connected and working; 3) switching between different operation modes; 4) demonstrating to a possible end users, such as KLM, the possibilities that a remote operator would have to control the robot.

In Fig. 5 (top), the main status page is shown which gives a quick overview if the robot's systems are working properly. In the middle, we see a map display that shows the robot's current position inside the airport, and its destination. Finally, on the bottom, we see a page that allows the remote operator to view which task the robot is currently performing, along with the sub-task and task-specific information (e.g. names of the guided passengers). Also, the main operation mode can be changed or reset here, which will cause a switch to a different high-level SMACH state machine, and a subsequent reset of the GUI.

3.3.4 Boarding Pass Reader Integration

A commercial boarding pass scanner was installed on the robot along with its software. The scanner software is able to output a text string containing all the information encoded in the barcode of a passenger's boarding pass.

We created a ROS node (boarding_pass_parser) that takes as input the string produced by the scanner and parses it according to the bar-coded boarding pass standard (BCBP) that is documented in the International Air Transport Association's (IATA) resolution 792 [1]. Individual information fields such as the passenger's last name and flight number were extracted and passed on to other nodes as a

[1] http://www.iata.org/whatwedo/stb/documents/bcbp_implementation_guidev4_jun2009.pdf

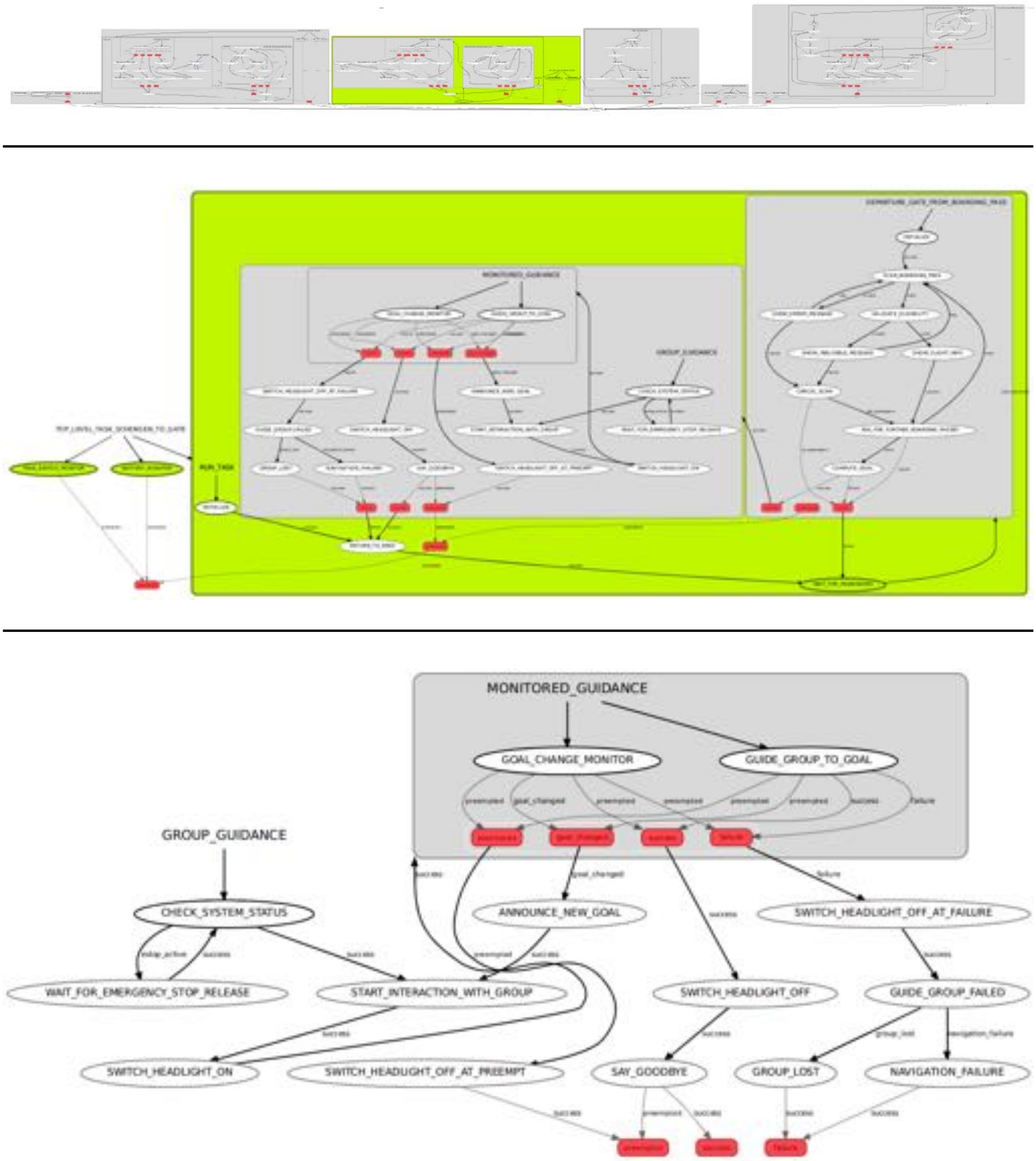

Figure 4: SMACH-based state machines for task execution, visualized using the built-in smach_viewer. *Top:* Overview of the 6 high-level state machines used during the final deployment at Schiphol. *Middle:* A single high-level state machine, representing one operation mode (here: Schengen-to-Gate guidance). *Bottom:* Digging even deeper into the state machine hierarchy, this state machine represents the group guidance task (used in multiple higher-level state machines, e.g. Schengen-to-Gate and Gate-to-Gate mode). The actual group guidance supervisor, a separate ROS component that ensures e.g. that the group is still following and maintaining a consistent distance, is represented by the 'guide group to goal' state in the grey box. All other states are mainly used to interface with the graphical user interface.

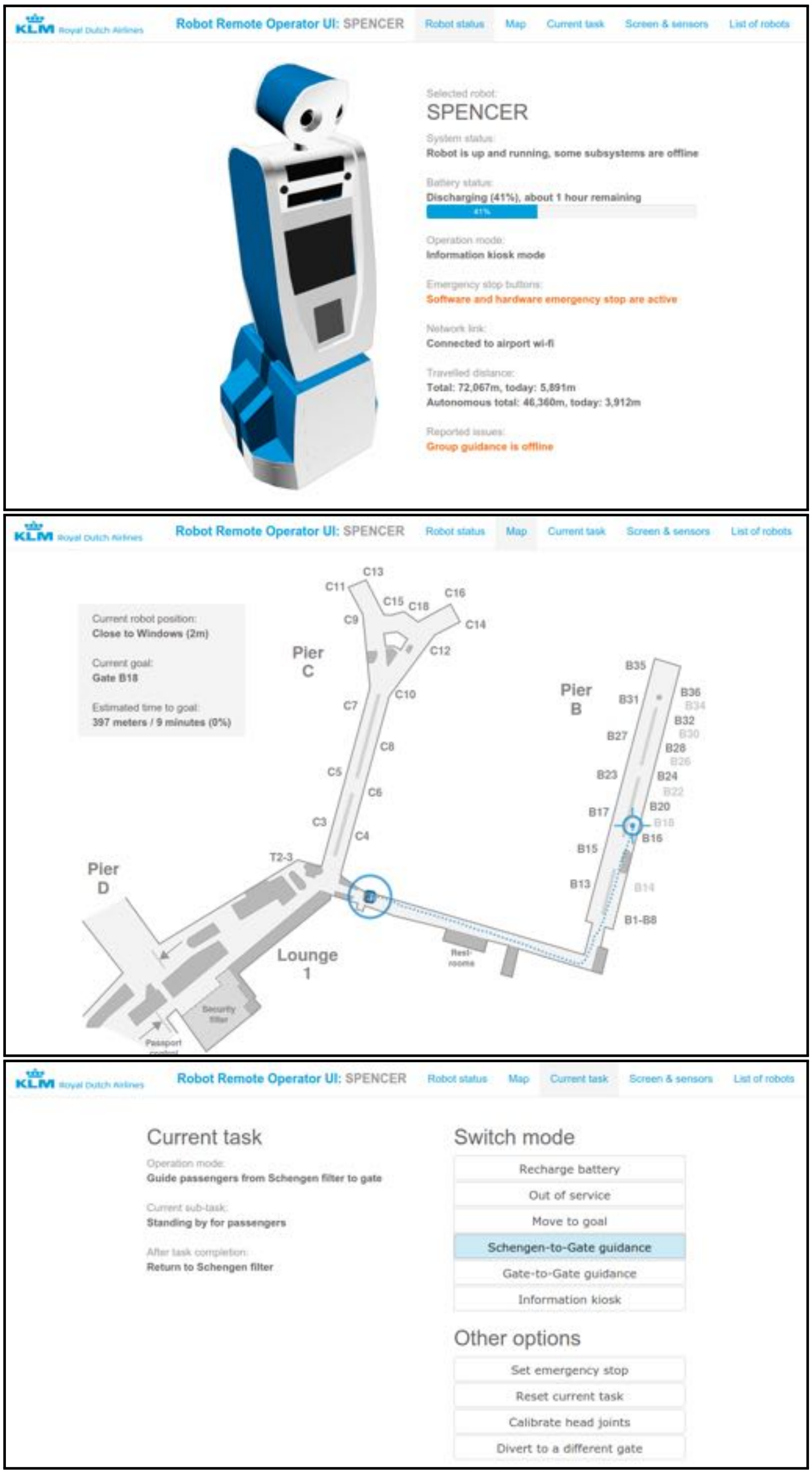

Figure 5: Web browser-based remote operator user interface, for diagnostic purposes and switching between different operation modes.

custom ROS message.

As the BCBP protocol does not enforce the gate number to be encoded in the boarding pass barcode and the gate number can change after a passenger has passed through the check-in and security control, we had to find another way to determine the target gate. For this purpose, we collaborated with our partner, KLM. The goal was to create a database with flights and gates information that the robot could access in real-time. Our solution was based on e-mail. We installed a light-weight e-mail client on the robot specifically for this task. The workflow was the following: The airline sends an e-mail to the robot once a day with all current flight-to-gate information. Whenever there is a gate change, the airline sends another e-mail. The client on the robot is configured to poll every minute for new emails and updates its internal database accordingly. Whenever a passenger scans his ticket, the robot can immediately query the database to find the target gate based on the flight number.

3.4 Mapping and Localization

After the tests during integration week IVb it became apparent that for bullet-proof localization in a very crowded environment such as Schiphol it was necessary to modify the sensor setup. Too many times, 2D laser range finders on the robot get fully occluded by moving people for extended periods in time and cause localization to become overly uncertain. Thus, for the final demonstration we have retrofitted the platform with a Velodyne VLP-16 3D sensor for localization and mapping purposes, mounted about 180 cm above ground. The mounting height helped to avoid occlusions and allowed to perceive more static parts of the environment. After a having resolved a last technical issue in the alignment of the maps used for planning and localization, this has ensured highly robust and accurate localization during the entire deployment of the SPENCER robot at Schiphol.

The final system was using a 3D map of of the deployment area (Fig 6). To reduce computation time we have combined two methods. First we discard all measurements below a certain height. In this way, we omit those parts of the scan data that are most likely caused by people and other dynamic objects (in an airport, we can safely assume planar motion). We also subsample the set of distributions in the NDT representation of the scan data, thus matching a sparse representation of the current scan to the map.

3.5 People Analysis

In addition to the human detection and tracking components of the SPENCER architecture, we have fully integrated the close-range people analysis modules head pose, body pose and upper body skeleton estimation. All three components were demonstrated at the end of the final integration week.

The components take as input the tracks from a second RGB-D tracker which runs in parallel to the multi-modal tracking system. The reason for the two trackers running in parallel is that the multi-modal tracking system gurantees a 360° coverage due to the laser sensors for a save navigation with reliable people tracking, while the RGB-D tracker provides more reliable bounding boxes in camera space, which gives a more precise input for the analysis modules, operating on the images of the camera space.

The head and body pose estimation was trained on data which had also been recorded during the integration meeting. Both were tested in the challenging environment of the airport and yielded

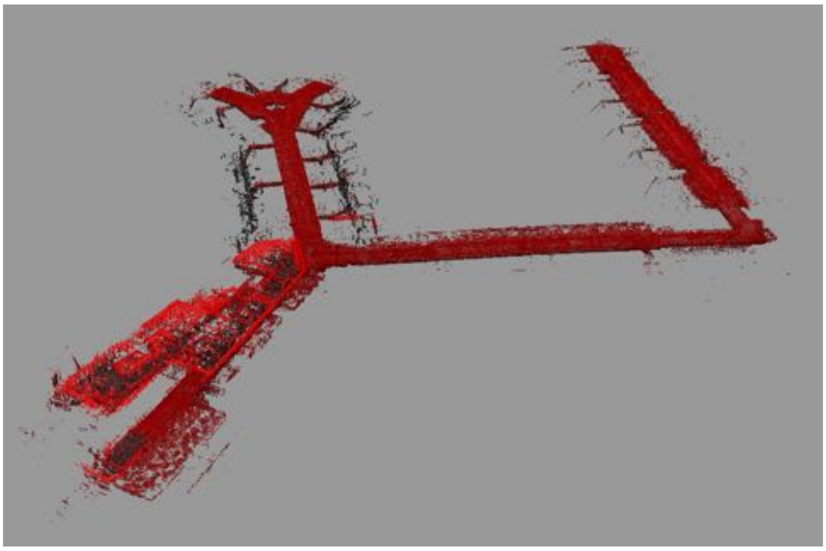

Figure 6: 3D-NDT map of the test area.

qualitatively nice results. Especially the head pose estimation provides useful information such as, if a person has seen the robot or looks at the robot right now. We also learned some interesting lessons while employing the skeleteon detection during the integration meeting. Here, the usual behaviour of the people was a standard walking pattern, which was nicely detected by our component in most cases. We also performed some first experiments on interaction gestures, i.e., a person reaching out with his arm to interact with the robot. Although not trained on complex gestures, it was still possible to detect the skeleton to some extent, posing interesting follow-up research directions.

An additional experiment was to re-use the estimated people analysis back in the tracking system. Here, the input trajectories have been refined using the estimated body poses as current orientation of the persons. For now, this mainly serves for visualization purposes, but further research is planed with the help of the recorded datasets at the airport, providing very helpful and challenging scenarios.

4 Visual Impressions

Finally, we give visual impressions of the final integration week and demonstration (Fig. 8) as well as screen shots of the professionally made video (Fig. 7).

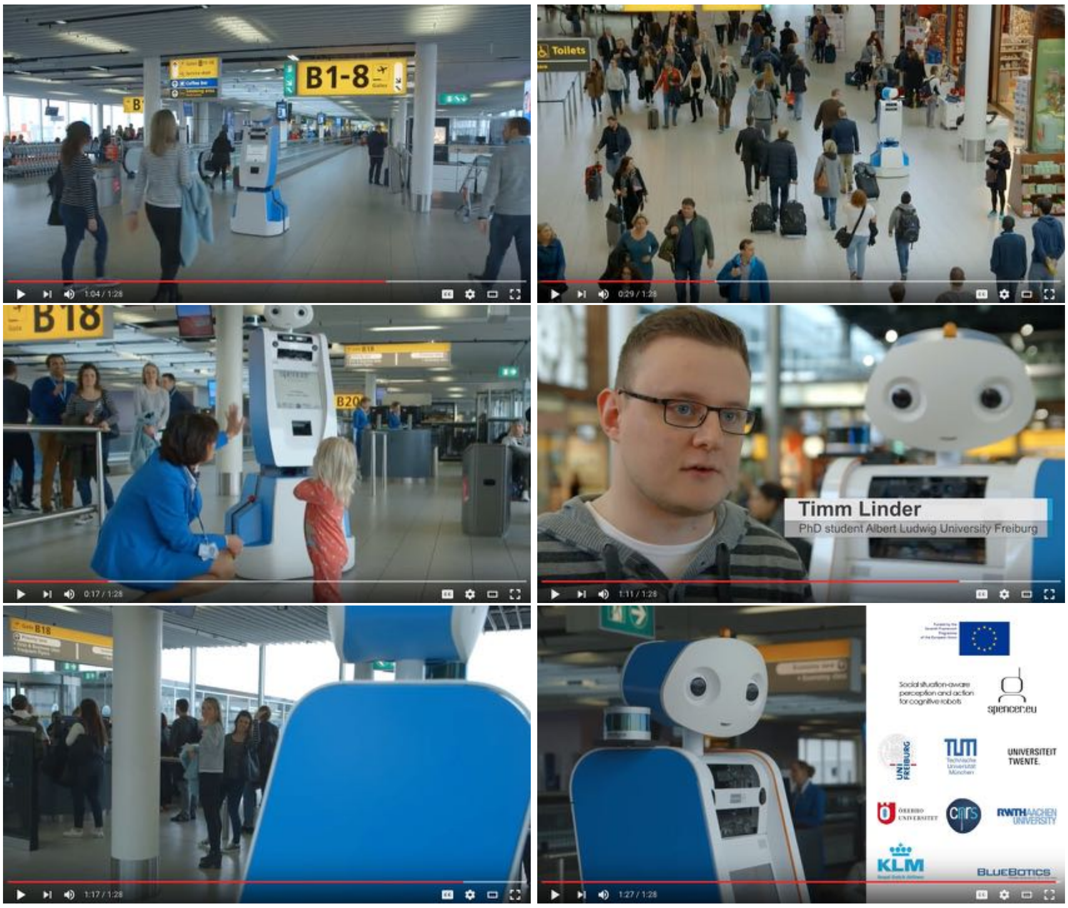

Figure 7: Example frames from the professional SPENCER presentation video shot at Schiphol Airport. By the end of May 2016, the video has over 19,000 views and 114 likes on YouTube.

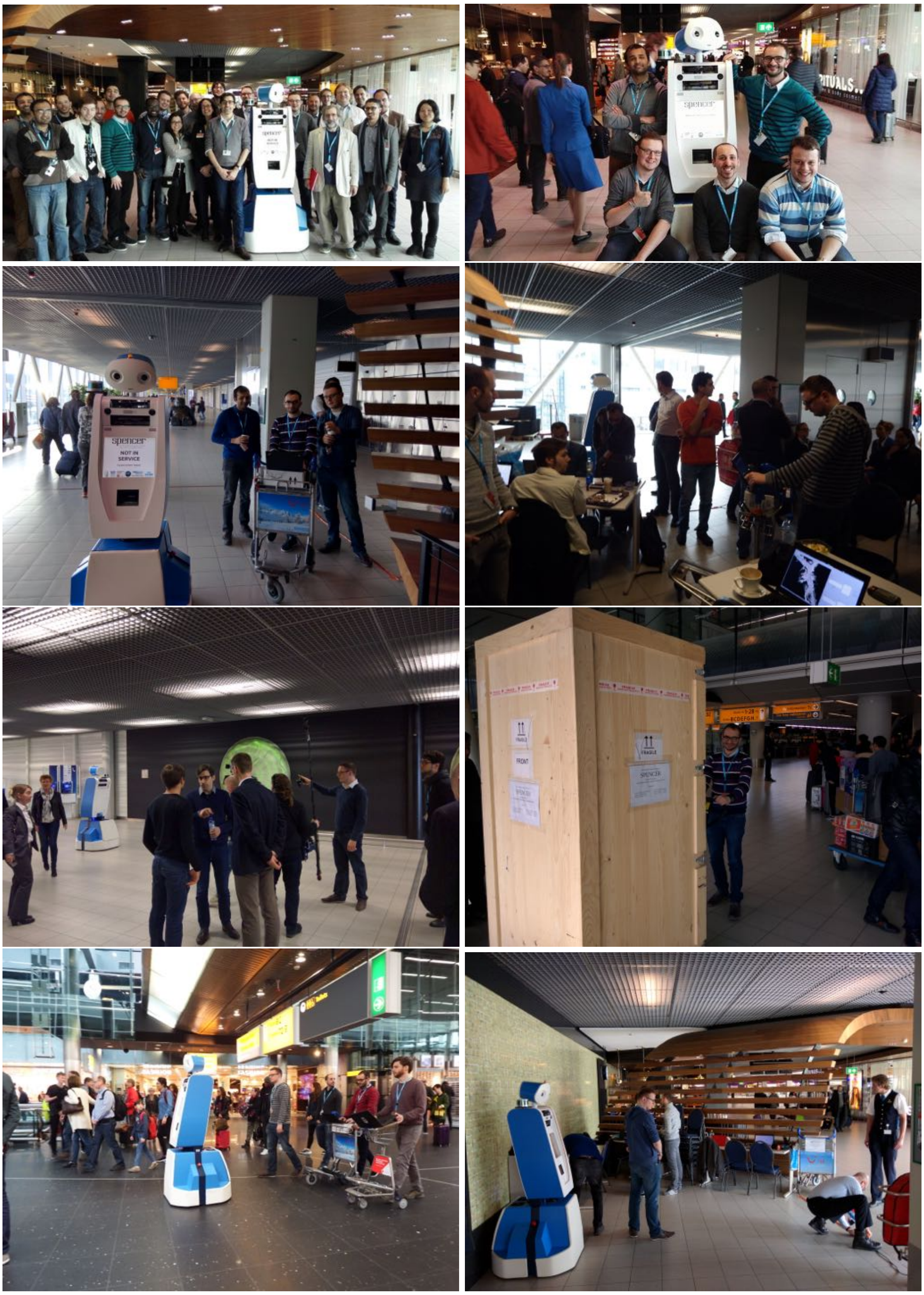

Figure 8: Impressions from the final SPENCER integration week and deployment.