

# Lindsey the Tour Guide Robot - Usage Patterns in a Museum Long-Term Deployment

Francesco Del Duchetto, Paul Baxter and Marc Hanheide

**Abstract**—The long-term deployment of autonomous robots co-located with humans in real-world scenarios remains a challenging problem. In this paper, we present the “Lindsey” tour guide robot system in which we attempt to increase the social capability of current state-of-the-art robotic technologies. The robot is currently deployed at a museum displaying local archaeology where it is providing guided tours and information to visitors. The robot is operating autonomously daily, navigating around the museum and engaging with the public, with on-site assistance from roboticists only in cases of hardware/software malfunctions. In a deployment lasting seven months up to now, it has travelled nearly 300km and has delivered more than 2300 guided tours. First, we describe the robot framework and the management interfaces implemented. We then analyse the data collected up to now with the goal of understanding and modelling the visitors’ behavior in terms of their engagement with the technology. These data suggest that while short-term engagement is readily gained, continued engagement with the robot tour guide is likely to require more refined and robust socially interactive behaviours. The deployed system presents us with an opportunity to empirically address these issues.

**Index Terms**—service robots, long-term autonomy, human-robot interactions

## I. INTRODUCTION

Deploying robots in public spaces to behave autonomously requires a number of challenges to be overcome, especially if the robot is intended to interact socially with the people therein. In this paper we present *Lindsey*, an autonomous robot deployed at The Collection museum<sup>1</sup> in Lincoln, UK (see Fig. 1). Lindsey started operating at the museum in October 2018 as an autonomous tour guide for visitors, and is planned to be functional for up to three years. The museum is open daily from 10 AM to 4 PM and during this time Lindsey goes around the archaeological section of the gallery providing guided tours and other information about the exhibitions to the visitors. Every morning, at opening time, the robot leaves its charging station and goes around looking for visitors to interact with. When the museum closes, it autonomously returns to his base station for charging the battery and uploading on the server the data collected during the shift. Lindsey can provide information about some specific exhibitions in the gallery, guide the visitors to the place the exhibition is located or perform entire guided tours.

The robot framework is designed to maximize autonomy during its operations. Since its initial deployment, Lindsey has



Fig. 1: (left) Lindsey at its base station. (right) students interacting with Lindsey, using the mounted touchscreen.

travelled autonomously almost 300 km and delivered more than 2300 guided tours, with almost no assistance from us.

The main contributions of this paper are:

- the proposal of Petri net plans (PNP) as a formal language to define robust behaviors in a declarative way, facilitating learning and adaptation;
- the description of the implemented interfaces to allow remote monitoring and management of the robot to the roboticists and the staff of the museum;
- an analysis of the engagement of users from 7 months of autonomous deployment of the system in a real-world museum context;
- a discussion of encountered challenges and lessons learned that can inform similar applications and future developments of the on-going project.

This paper reports the software architecture of the robot (Section III) and the behaviours it is endowed with (Section IV), the initial results in terms of autonomy and public engagement (Section V), and discusses future directions to give the existing system the ability of learning, from the interactions with the visitors, to provide more engaging guided tours (Section VI).

## II. RELATED WORK

In the past, several works have addressed the problem of Long-Term Autonomy (LTA) in robotics. Meeussen et al. [1] have deployed a robot for 13 continuous days in an office environment, while exploring ways for improving the robot robustness by identifying failures and recoveries (including

F. Del Duchetto, P. Baxter and M. Hanheide are with the L-CAS, University of Lincoln, UK. {fdelduchetto, pbaxter, mhanheide}@lincoln.ac.uk

<sup>1</sup><https://www.thecollectionmuseum.com/>

asking for help to humans). In [2] a fleet of four CoBots reached 1,000 km of overall autonomous navigation. The robots were able to seek human assistance to perform manipulation tasks (the robots did not have arms) and send emails to developers in case of lack of human response. Within the STRANDS project [3], the SCITOS G5 robot travelled more than 160 km over three individual deployments. Even though the robot was able of being autonomous for most of the time, the authors report the need for a way to manage failures and to have a better understanding of human activities. [4] propose a spatio-temporal model to learn when, where and how users interacted with the robot info-terminal during a long-term deployment. They found they could improve efficiency and usefulness of the system proposing the right content at the right time and place. Building from these works, we designed a system that is able to operate autonomously in a public environment for months while interacting with humans.

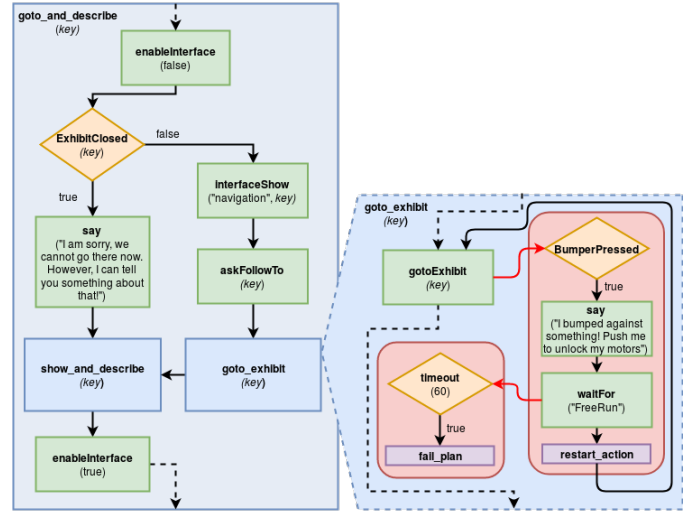
A survey on long-term interaction between users and robots [5] raises the issue that memory and adaptation remains nearly unexplored in the field. Similarly, [6] explores the state of the art on Artificial Intelligence (AI) techniques for LTA asserting that a major future challenge is that of integrating human interactions in the robot system to allow improving its knowledge in unforeseen situations. With the software architecture described in this work, we lay the foundations for a robot framework that allows exploiting the human feedback during interaction in order to optimize its social abilities.

Previous works featured a robot deployed in a museum environment. The robot Rhino [7] was deployed in a museum in Germany for 6 days guiding hundreds of visitors. At the time, the main issues and the focuses of the work were navigation and obstacle avoidance. The Minerva robot [8] traversed more than 44 km and interacted with more than 50k people. Moreover, it was able to display mood (i.e. happy or angry) and used an RL approach to learn the best actions to engage visitors. In [9] four robots were deployed over five years. Focusing on interactivity and education they learned that: 1) usually there is a crowd around the robot therefore any speech from the robot should be multimodal; 2) long and non-interactive presentations are guaranteed to drive audience away.

In the present work, we plan to address these issues by enabling the robot with the ability to directly optimize the users' engagement during the interaction. The first stage in doing so is characterising the engagement with the robot.

### III. ROBOT FRAMEWORK

Lindsey is a Scitos G5 robot manufactured by MetraLabs GmbH. In order to sense the environment, the robot has a laser scanner with 270° scan angle on its base and an Asus xtion depth camera mounted on a pan-tilt unit above his head. The interactions with the visitors are mediated through a touch screen, two speakers, a microphone and a head with two eyes that can move with five degrees of freedom to provide human-like expressions. To ensure safe operations in public environments the robot is equipped with an array of bumpers around the circular base with sensors to detect collisions and



**Fig. 2:** Block diagram to represent visually the plan `goto_and_describe`, with the expansion of the sub-plan `goto_exhibit`. Green nodes are low level actions, orange nodes are conditions, light-blue nodes are plans and purple nodes are ER statements. Red containers enclose ER components. Arrows shows connections between actions/sub-plans, conditions and statements. Black arrows coming from actions/sub-plans are only traveled after the source action/sub-plan is executed. Red arrows are connection that are active while the source action is in execution. Values between round brackets are the parameters passed to the action/condition/plan.

two easily reachable emergency buttons that, when activated, cuts the power to the motors.

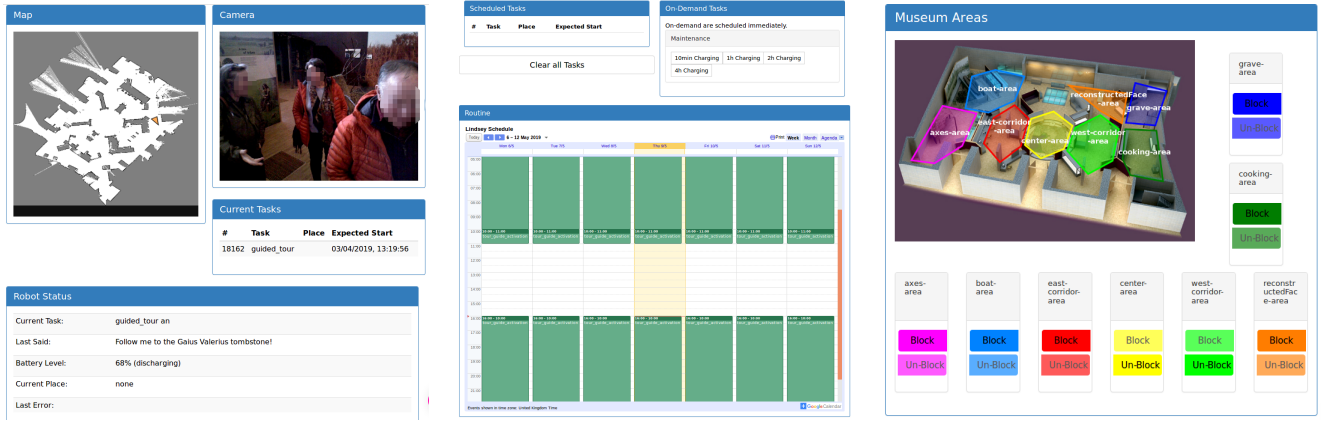
The software framework is based on ROS and uses STRANDS project [3] core modules for topological navigation, people tracking, task scheduling and data collection.

#### A. Robot behaviours specification

The robot behaviors are specified through tasks which represent the high-level activities of the robot. Each task has an associated priority, duration and a time window for execution. Whenever a task is demanded a scheduler takes care of deciding if and when it must be executed, considering the other tasks already in the schedule. Higher priority tasks take precedence over tasks with lower priority at this stage.

The tasks that require the robot to perform complex behaviors, such as interacting with people, are specified as conditional plans using the Petri net plans (PNP) formalism [10]. The PNP formalism allows to define conditional plans that can be directly translated into a stochastic policy allowing learning and adaptation [11]. Moreover, the PNP library provides tools for executing and monitoring the plans.

A conditional plan is specified as a sequence of actions and conditions. An action corresponds to the implementation of a low-level capability that the robot system can execute. Actions can be started, stopped and they modify the state of the environment and/or the internal state of the robot with their execution. A condition, instead, represents the truth value of a particular feature of the environment or of the robot system. Conditions can be evaluated at every moment during the exe-



**Fig. 3:** Screen-shots of the management web interface. From left to right: the home page shows the position of the robot, the camera live-feed and some other information about the robot state; the tasks page allows to monitor the currently scheduled tasks and the routines that managed by the calendar, with the possibility of clearing all the tasks scheduled or temporarily sending the robot to the charging station; tool for blocking areas of the museum wherein the robot can not navigate.

cution of a plan. Furthermore, PNP allows to interrupt actions in the plan at any stage of their execution with the use of *execution rules* (ER). ER are tuples represented as  $(a, \phi, \sigma, \rho)$  where  $a$  is an action,  $\phi$  is a boolean expression,  $\sigma$  is a program (an action, a sequence of actions) and  $\rho \in \{\text{restart\_plan}, \text{restart\_action}, \text{skip\_action}, \text{fail\_plan}\}$  is a statement which determines how to continue the execution of the plan. Whenever  $\phi$  is satisfied during the execution of  $a$ , the action is stopped and the program  $\sigma$  is executed.  $\rho$  determines how to resume with the plan afterwards.

In our implementation, each plan is also a sub-plan. Therefore, a plan can be called from other plans, in the same manner an action can be called. This feature allows to specify the robot behaviours through hierarchical plans and allows to reuse the single components of the plans. For ease of understandability, we show, in Figure 2, a block diagram representing the plan `goto_and_describe`.

A straightforward example, showing how the use of PNP can facilitate writing robust behaviors, is that of specifying a plan that can deal with the unforeseeable situations that can happen during navigation. When the robot is travelling to an exhibit, the following events can prevent the robot from progressing with the execution of the plan if not properly handled with recovery actions:

- *emergency button pressed*: pressing an emergency button on the robot cuts the current to the motors, making the robot unable to travel. To re-activate the robot motors the emergency button must be manually released. When they are pressed the robot actively asks the users to release them when a navigation action is executed.
- *bumper pressed*: when the bumper on the base of the robot detects a collision a software node blocks the robot motors. The motors block can be re-activated on-demand by software. After a collision is detected, and the motors are blocked, the robot asks the users to be slightly pushed to signal it can continue to navigate. Therefore, if a push is detected (as a change in the robot's odometry) the motors block is released.

- *navigation failure*: the navigation planner is not able to generate a viable path in the topological map because the robot is stuck close to some obstacles. In this case, the robot immediately stops its navigation action and interactively asks the user to be dragged away from the obstacle. After being moved to a zone clear from obstacles it restarts the navigation action.

In these cases, Execution Rules (ER) are used for correctly handling the recoveries and to ensure the successful plan progression. As shown in Figure 2, the sub-plan `goto_exhibit` handles the situation of the bumper being pressed with an ER which temporarily interrupts the execution of the action `gotoExhibit` to interactively ask the user's help. The ER returns the execution to the action interrupted only when the plan progression is ensured (i.e. the user unlocked the motor software block by pushing the robot), otherwise it fails the plan (i.e. after waiting for help with no success for more than a minute).

### B. Management interface and remote monitoring

Lindsey can be monitored and partially managed by the museum employees through a web application. The application is served over the internet from a computer located in the collection, connected to the same subnetwork the robot is connected to. Likewise the user interface, the management interface communicates with the robot ROS framework through the `roslibjs` library.

Figure 3 shows some screen-shots of the interface. The *Status* page shows the feed from the robot camera, the robot pose on the museum map, the tasks in execution and other information about the robot state (e.g. the last sentence said, the battery level and the topological position). The *Tasks* page allows to send the robot to the charging station for some time (e.g. during a malfunction), it shows the robot calendar with the routine behaviors and it allows to block or unblock the museum areas. When a museum area is blocked the robot is not allowed to travel to locations within that area. Therefore, it will plan paths around it when possible or, otherwise, abort the navigation action.



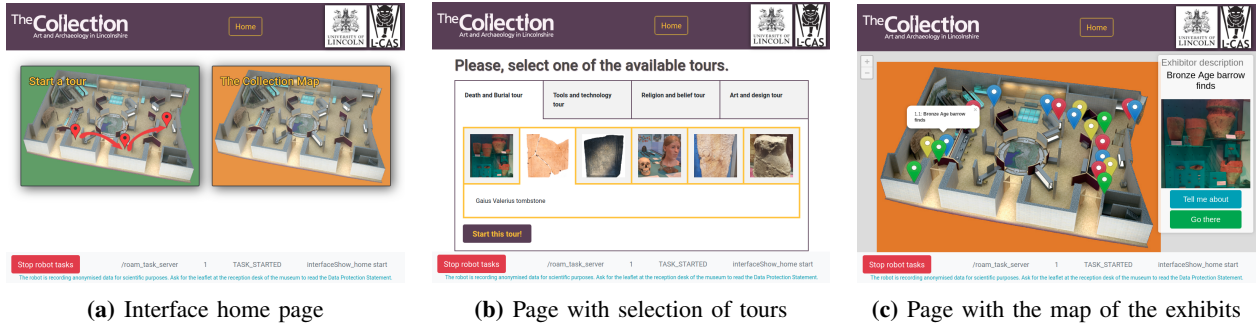


Fig. 4: Screenshots of the robot graphical interface shown on the robot tablet.

### C. Critical events notification

During the long term deployment of an autonomous robot, operating without expert human supervision, it is important that the roboticists are promptly informed whenever a critical situation occurs. To this aim, we have designed a ROS module<sup>2</sup> that allows to define conditions over the robot internal state that are continuously checked during the robot operations. Each time a condition is satisfied the module issues a message, that is sent to a Slack<sup>3</sup> channel, containing the value of the robot state along with a screenshot from the robot camera feed.

This mechanism gives us the possibility to react quickly in situations that can halt the normal system operations and that are possibly damaging for the visitors, like a sensor malfunction or the battery discharging up to a critical value.

### D. User interface

The user interface is a web application implemented in HTML/CSS and JavaScript which is displayed by the touch-screen mounted on the robot. It communicates with ROS through roslibjs. The interface is the primary mean of interaction between Lindsey and the museum visitors. The user can browse the exhibits of the archaeological collection displayed on an interactive map and request the robot to provide some information about it or to be guided to the place it is located. Moreover, the user can visualize the available robotics tours, with a preview of the exhibits featured in it, and decide to start one of them. Whenever the user demands to get information about an artifact, to be guided to an exhibit or to start a guided tour the robot takes control of the interface showing contextual information about the behaviour it is performing.

The interface is therefore controlled by two parts, the museum visitors and the robot. Sometimes one of the parts has the exclusive control over the interface, other times the control is shared by the two. Ultimately, the robot has the potential to decide at any moment to act on the interface (regardless of whether the visitors are using it or not) and to disable the interface for the visitors. In the latter case, all the handlers of the interface touch events are disabled, except the handler that stops the robot task in execution (which remains always active).

Figure 4 shows some pages of the web interface. The home page (Fig. 4a) allows the visitors to navigate to the tours page

and to the map page. The tours page (Fig. 4b) lists the available guided tours, each featuring 5 or 6 different artifacts present in the gallery. The map page (Fig. 4c), instead, shows a static map of the archaeological gallery with all the artifacts known by the robot. For each artifact, the user can ask to be guided by the robot to its location and to receive a short description. The footer displays the state of the robot execution by showing the task, its state, its priority and the current action of the plan. Moreover, it allows the user to stop, at any moment, a task during its execution.

## IV. ROBOT OPERATIONS

The robot operations are specified through tasks and their execution is managed by the scheduler, as explained in Section III-A. In this section, we focus in particular on describing the tasks that Lindsey executes in our museum deployment.

### A. Routine robot operations

The museum is open to the public 7 days per week from 10:00 AM to 4:00 PM. Every morning, at opening time, the robot executes the *Tour guide activation* task which gives instruction to autonomously leave the charging station, to activate the user interface and to start the roaming behavior. Similarly, at closing time the *Tour guide deactivation* task is started which instructs the robot to return to the charging station, deactivate the interface, stop the roaming behavior and upload all the data collected during the shift on the cloud.

The *Tour guide activation/deactivation* tasks are defined through the calendar utility and therefore automatically scheduled every day.

### B. Roaming

The goal of this behavior is to get the robot to move around the gallery, whenever it is not executing any other task, in order to be available to the visitors where and when they most need it. The robot will move in those areas where people are expected to be with a higher probability at that specific time, while continuously adapting its internal spatio-temporal model. For more information about the algorithm for predicting where the robot should go the reader is redirected to [4].

A ROS node submits a low-priority roaming task every 5 minutes. The task is a plan which shows the home page on the robot screen and invites people to interact with it, whenever it detects one nearby.

<sup>2</sup><https://github.com/LCAS/sentor>

<sup>3</sup><https://slack.com>

Days of operation	103 days
Total distance travelled	299 km
Total tasks completed	8423
TLS	26 days, 11 hours
A%	74%

**TABLE I:** Long-Term Autonomy metrics.

Task	Tot. demanded	Average duration	Median duration	Shortest	Longest
<i>Guided tour</i>	2367	4.52 [min]	3.13 [min]	11 [sec]	22.15 [min]
<i>Go to exhibit and describe</i>	2379	1.87 [min]	1.84 [min]	8 [sec]	30.79 [min]
<i>Describe exhibit</i>	486	26.73 [sec]	20.03 [sec]	7.31 [sec]	5.78 [min]

**TABLE II:** Number of user demanded tasks with their duration.

### C. User Demanded Tasks

The following behaviours are demanded by the visitors of the museum through the interaction with the user interface.

1) *Guided tour*: A tour is composed of a sequence of exhibits linked by a common theme. The robot initially gives a description (verbally and by showing images on the touchscreen) of the tour providing some context for the exhibits then guides the visitors to each stop of the tour sequentially providing a description. The robot initially gives some brief information about the exhibit, then asks the visitors if they want to know more. The visitors can reply with the touchscreen through a yes/no modal window or by verbally pronouncing their answer. In case of a negative answer, the robot resumes the tour guiding the visitors to the next stop. In case the robot does not receive an answer within 1 minute it terminates the tour, assuming that the visitors left. Whenever one or more stops of the tour are not available, because the area in which they are located is closed, the robot goes first to those that are available and then describes the remaining ones without guiding the visitors to their location. In this case, the robot informs the user of the area blockage.

2) *Go to exhibit and describe*: The robot guides the visitors to an exhibit of their choice and then describes it (verbally and by showing images on the touchscreen). As for the *Guided tour* stops description, the robot offers to provide additional information to the visitors and it does so only in case they agree. If the robot is forbidden to travel to the area in which the exhibit is located, it describes the exhibit without reaching the location (informing the user of the location blockage).

3) *Describe exhibit*: The robot gives a short verbal description of the exhibit demanded by the visitor.

## V. DATA ANALYSIS

The data shown in this section refers to the date range between the 24<sup>th</sup> January 2019 (day on which we started recording data of the robot operations) and the 9<sup>th</sup> May 2019, despite the robot being operative in the museum since October 2018. To assess the robot performance in terms of Long-Term Autonomy (LTA) during the current deployment we report the overall system performance against two metrics: *total system lifetime* (TSL), and *autonomy percentage* (A%), as previously done in [3]. TSL measures how long the system is available for

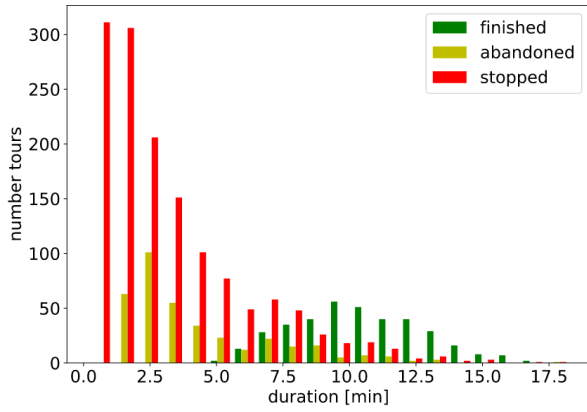
autonomous operation, A% measures the duration the system was actively performing tasks as a proportion of the time it was allowed to operate autonomously (which in our case is restricted to the museum opening hours). Table I shows these metrics along with other LTA measures.

To evaluate the performance of our robot in terms of public engagement we analyse the user demanded tasks. Although we do not directly measure the public engagement level with the robot, we estimate it by studying the robot usage which is a component of user engagement and a more accessible value to measure.

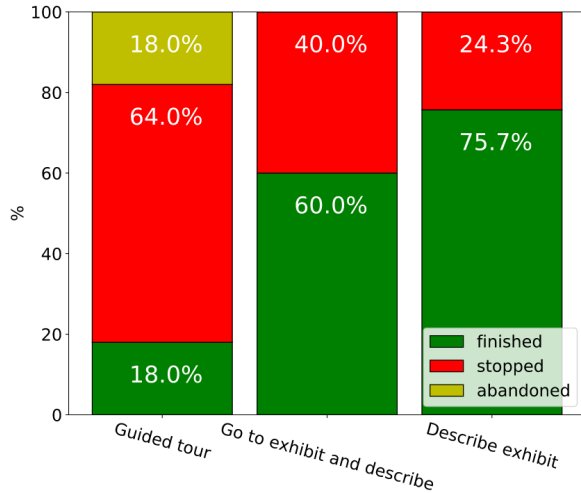
Table II shows, for each of the task classes, the total number of tasks demanded by the visitors alongside their average, median, shortest and longest duration. As discussed in Section III-D, the interface allows the users to stop a task at any moment during its execution. The data presented in the table shows the overall number of tasks, regardless of how they ended. In fact, we can see that there is high variability in the duration of different instances of the same task. A normal *Describe exhibit* tasks typically lasted about 20 seconds, the shortest tasks are those stopped by the visitors while the longer ones were caused by a software bug. *Go to exhibit and describe* tasks normally lasts about 1.8 minutes. In this case, the duration of shortest tasks is caused by the visitors stopping them or by the fact that the robot did not navigate to the exhibit location which was closed (see Section IV-C). The longest tasks can be caused by software/hardware malfunctions or by the navigation action which is affected by unpredictable obstacles and failures. For the *Guided tour* task, we plot, in Figure 5, the duration of all the instances clustered according to how they terminated. Abandoned tasks are those stopped by the robot whenever it “thinks” that the visitors wandered off during the tour (as described in Section IV-C). A *Guided tour* task which is completed to the end (not stopped nor abandoned) has a duration of about 10 minutes.

A task that has a longer duration are stopped with higher probability than others, as we observe in Figure 6.

In Table III, we show the PNP actions, in the *Guided tour* task, in execution when the visitors stopped the task alongside with the associated number of occurrences. The robot tours are consistently stopped more often when the robot is navigating to an exhibit or when the robot is describing the exhibit.



**Fig. 5:** Histogram with the duration of *Guided tour* tasks clustered according to their finishing state. Here we have filtered out all the tours demanded when some of the stops location area was closed.



**Fig. 6:** Rate of tasks normally finished, stopped or abandoned by the visitors.

Figure 7 shows the total schedule of the tasks performed by the robot during the current deployment during opening hours of the museum. Given that the robot behaviours and capabilities have remained stable for most of the deployment, variations in the number of user demanded tasks over the different days depend on several factors that are out of our control, such as holidays, weather or other temporal exhibition displayed at the museum. For example, we observe a pattern where the tasks are more frequently demanded during week-ends. Moreover, the periods from the 16<sup>th</sup> February to the 23<sup>th</sup> February and from the 5<sup>th</sup> April to the 20<sup>th</sup> April, in which we observe a substantial increase in the number of user demanded tours with respect to the rest of the days, reflects school term holidays periods in Lincolnshire (i.e. Spring Half Term Holidays: 16<sup>th</sup> February - 25<sup>th</sup> February, Spring Holidays: 5<sup>th</sup> April - 23<sup>th</sup> April). White spaces in the robot schedules indicate that the robot is charging (typically before 10:00 AM and after 4:00 PM), idle (after a user demanded task) or that a failure has occurred.

Action	Stopped tours
GO_TO_EXHIBIT	496
DESCRIBE_EXHIBIT_ADDITIONAL	229
DESCRIBE_EXHIBIT	229
PROPOSE_ADDITIONAL_INFO	94
ASK_HELP BUMPER_PRESSED	93
DESCRIBE_TOUR	75
ASK_FOLLOW	74
NO_ACTION	51
ASK_HELP_EM	20
ASK_HELP_NAV_FAILURE	19
TOUR_ENDING_SENTENCE	18
THANKS_HELPED	2

**TABLE III:** Actions in execution when the user stopped the tour with associated number of occurrences.

## VI. DISCUSSION

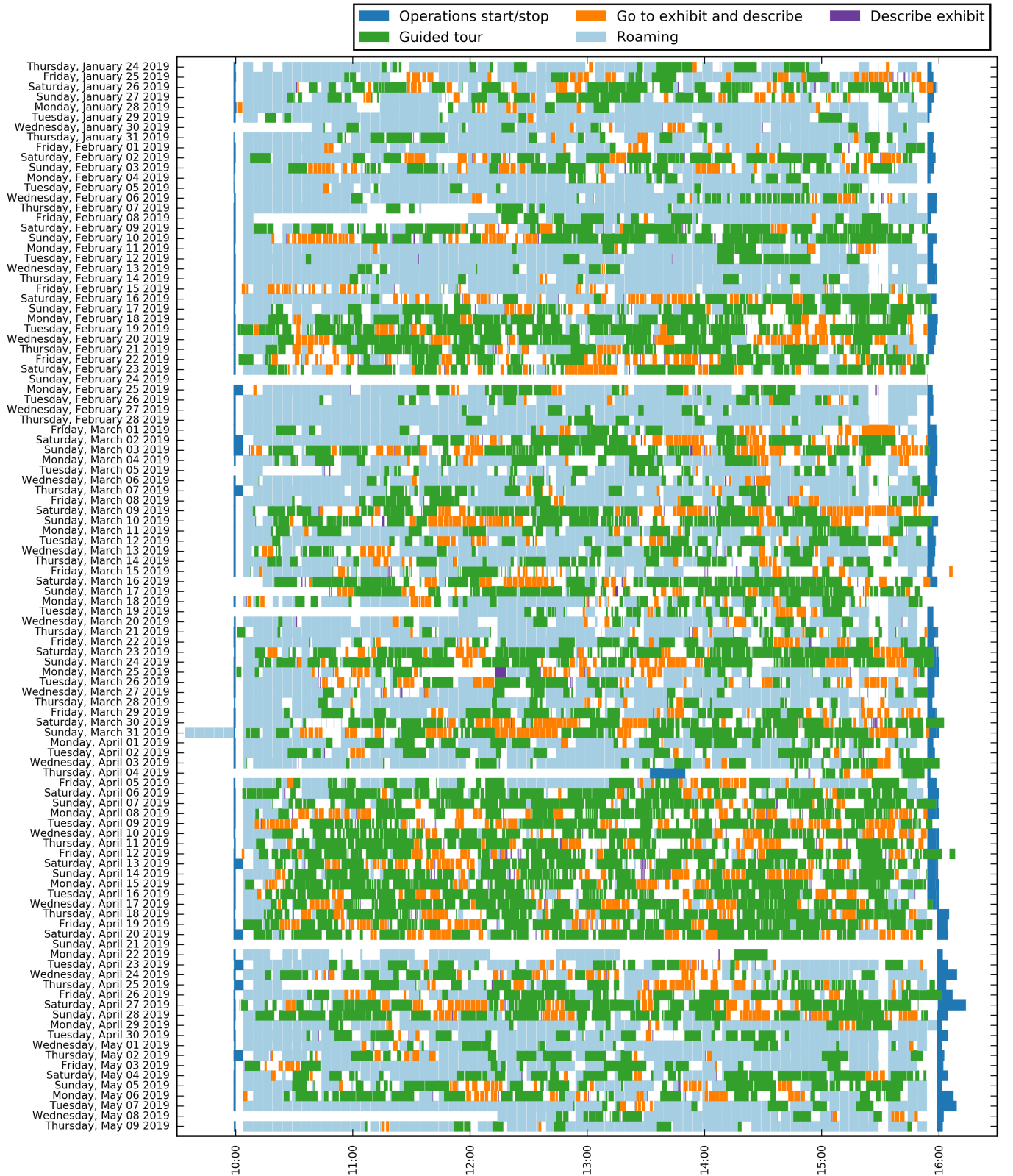
The large number of engagements started with the robot enables us to be confident in our observation that these engagements are soon lost during extended interaction, typically after around 2 minutes. This is consistent with prior results from similar deployments, where initial impressions of such robots are positive [12].

This result could mean either that  $\sim 2$  minutes is the average threshold of duration of human attention during guided tours, or that our robotic tours are not engaging enough to keep the visitors' attention alive for longer. The former hypothesis implies that this threshold of duration of attention would also be detected in human guided tours. Our experience would suggest that this is not the case. While in a typical human guided tour visitors cannot typically stop the tour (even when they want) and they would not wander off due to compliance with social norms (avoiding impoliteness toward the guide for example), in our robotic guided tours visitors are free to stop the tour whenever they like and they do not appear to feel the urge of complying with human social norms toward the guide. However, our intuition and experience tells us that although there can be very poor human guides that make you want to stop or leave a tour, this is a very rare occurrence.

The latter hypothesis, i.e. that the robot performs poorly in maintaining the engagement of the visitors during the tours, implies that the robot behaviors are not engaging enough. As with many robot deployments, there is a common (implicit) assumption that the robot is engaging by virtue of its presence alone. Our results are consistent with previous studies, such as [9], in suggesting that this effect is not sufficiently persistent.

From an alternative point of view, studies of human guided tours in museums have provided recommendations for how human guides can give better (more engaging) tours [13], such as:

- tours should not resemble monolithic lectures but they must be interactive;
- guides should facilitate audience contribution and engagement through questions and answers, also taking



**Fig. 7:** A plot of the tasks performed by the robot during deployment for nearly 4 months. White space indicates that the robot is not performing any tasks. This can indicate that the robot is idle (e.g. when the robot is charging) or that a failure has occurred. See Section IV for task descriptions.



into account non-verbal features like eye movements and posture;

- guides should seek to secure audience attention to inform and entertain them, encouraging them to orient to the feature under consideration;
- the audience should not be considerate as a whole but the guide must take into account features of the single people, even personalizing the experience;
- technologists need to create non-human guides that have a similar level of sensitivity to the audience built-in.

Although mostly aimed at human guides, the above list gives us useful insights for designing more engaging robotic guided tours. However, in order to fully implement the aforementioned recommendations there are a number of technological obstacles that must be overcome. These include the online detection of engagement, which is a substantial challenge, and the development of socially contingent robot behaviours to act on this information. This, in turn, suggests a range of technologies and techniques that are required to detect these features.

Deep learning approaches in recent years allowed substantial improvements in the fields of computer vision and natural language processing, however, it is still not clear how can we use those techniques in order to equip the robot with better social abilities. We propose to give the robot the ability to detect the engagement level of the people interacting with it. As previously done in [14]–[18], combining a set of features coming from perception, related to the humans in the robot's environment and to the context of the interaction, we can estimate the engagement of the participants in the interaction.

This can be used to enable the robot to learn, on-line from interactions, to select the actions that are expected to generate higher engagement for the humans interacting with it. This approach is supported by the long-term deployment in the museum, as the 'end users' would be directly involved in the learning process, and by the use of PNP to specify robot behaviors that can be easily translated into stochastic policies allowing learning.

## VII. CONCLUSION

In this paper we have introduced the "Lindsey" tour guide robot system which is deployed in a museum for an extended period of time to operate autonomously, acting as a guide to provide tours and information to visitors. So far, in a deployment lasting seven months up to now, it has travelled nearly 300km and has delivered more than 2300 guided tours. This platform has enabled us to systematically gather extensive data on how people interact with the robot as a guide. These data have indicated that beyond short-term engagement, further attention needs to be paid to increasing the social interactivity of the robot. While this constitutes ongoing work, the Lindsey robot presented here constitutes a stable interactive system that is deployed 'in the wild'.

## ACKNOWLEDGMENT

This work was jointly funded by EPSRC grant ("NCNR", EP/R02572X/1), EU H2020 ("ILIAD", 732737) and Lin-

colnshire County Council.

## REFERENCES

- [1] Wim Meeussen, Eitan Marder-Eppstein, Kevin Watts, and Brian P Gerkey. Long term autonomy in office environments. In *ALONE Workshop, In Proceedings of Robotics: Science and Systems (RSS11)*, Los Angeles, USA, 2011.
- [2] Joydeep Biswas and Manuela Veloso. The 1,000-km challenge: Insights and quantitative and qualitative results. *IEEE Intelligent Systems*, 31(3):86–96, 2016.
- [3] Nick Hawes, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrova, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Kortner, et al. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine*, 24(3):146–156, 2017.
- [4] Marc Hanheide, Denise Hebesberger, and Tomáš Krajník. The when, where, and how: An adaptive robotic info-terminal for care home residents. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 341–349. ACM, 2017.
- [5] Iolanda Leite, Carlos Martinho, and Ana Paiva. Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308, 2013.
- [6] Lars Kunze, Nick Hawes, Tom Duckett, Marc Hanheide, and Tomáš Krajník. Artificial intelligence for long-term robot autonomy: A survey. *IEEE Robotics and Automation Letters*, 3(4):4023–4030, 2018.
- [7] Wolfram Burgard, Armin B Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The interactive museum tour-guide robot. In *Aaai/iaai*, pages 11–18, 1998.
- [8] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Hähnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 3. IEEE, 1999.
- [9] Illah R Nourbakhsh, Clayton Kunz, and Thomas Willeke. The mobot museum robot installations: A five year experiment. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 4, pages 3636–3641. IEEE, 2003.
- [10] Vittorio A Ziparo, Luca Iocchi, Daniele Nardi, Pier Francesco Palamara, and Hugo Costelha. Petri net plans: a formal model for representation and execution of multi-robot plans. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 79–86. International Foundation for Autonomous Agents and Multiagent Systems, 2008.
- [11] Matteo Leonetti and Luca Iocchi. Learnnpnp: A tool for learning agent behaviors. In *Robot Soccer World Cup*, pages 418–429. Springer, 2010.
- [12] Daphne Karreman, Geke Ludden, and Vanessa Evers. Visiting cultural heritage with a tour guide robot: A user evaluation study in-the-wild. In *ICSR 2015*, pages 317–326. Springer LNAI, 2015.
- [13] Katie Best. Making museum tours better: Understanding what a guided tour really is and what a tour guide really does. *Museum Management and Curatorship*, 27(1):35–52, 2012.
- [14] Candace L Sidner, Christopher Lee, Cory D Kidd, Neal Lesh, and Charles Rich. Explorations in engagement for humans and robots. *Artificial Intelligence*, 166(1-2):140–164, 2005.
- [15] Séverin Lemaignan, Fernando Garcia, Alexis Jacq, and Pierre Dillenbourg. From real-time attention assessment to with-me-ness in human-robot interaction. In *The eleventh acm/ieee international conference on human robot interaction*, pages 157–164. IEEE Press, 2016.
- [16] Marek P Michalowski, Selma Sabanovic, and Reid Simmons. A spatial model of engagement for a social robot. In *9th IEEE International Workshop on Advanced Motion Control, 2006.*, pages 762–767. IEEE, 2006.
- [17] Aaron Holroyd. Generating engagement behaviors in human-robot interaction. 2011.
- [18] Mary Ellen Foster, Andre Gaschler, and Manuel Giuliani. Automatically classifying user engagement for dynamic multi-party human-robot interaction. *International Journal of Social Robotics*, 9(5):659–674, 2017.