

# Building Proactive Voice Assistants: When and How (not) to Interact

O. Miksik<sup>1\*</sup>, I. Munasinghe<sup>1\*</sup>, J. Asensio-Cubero<sup>1\*</sup>, S. Reddy Bethi<sup>1\*</sup>, S-T. Huang<sup>1</sup>, S. Zylfo<sup>1</sup>, X. Liu<sup>1</sup>, T. Nica<sup>1</sup>, A. Mitrocsak<sup>1</sup>, S. Mezza<sup>1</sup>, R. Beard<sup>1</sup>, R. Shi<sup>1</sup>, R. Ng<sup>1</sup>, P. Mediano<sup>1</sup>, Z. Fountas<sup>1</sup>, S-H. Lee<sup>1</sup>, J. Medvesek<sup>1</sup>, H. Zhuang<sup>1</sup>, Y. Rogers<sup>2\*</sup>, P. Swietojanski<sup>1\*</sup>

<sup>1</sup>Work done at Emotech Labs, U.K., <sup>2</sup> UCL Interaction Centre, University College London, U.K.

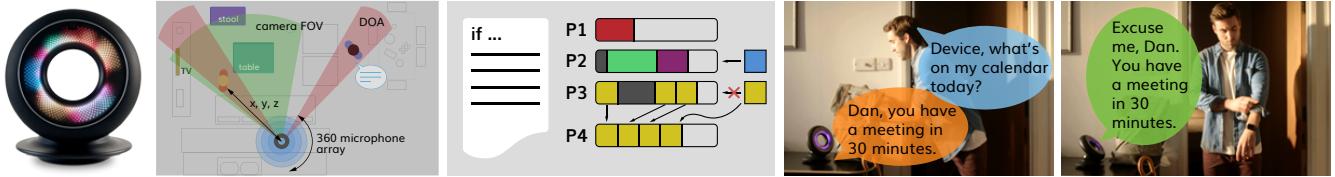


Figure 1: We propose that the next generation of voice assistants should be capable of proactive-initiated interactions. We discuss design principles and validate them using a (a) new hardware platform equipped with (b) multi-modal semantic scene understanding and (c) decision making modules. Using the proposed design assumptions, hardware and software, we demonstrate how users can benefit from transforming (d) reactive devices which typically **only respond requests initiated by the users**, to the (e) **proactive ones** offering the user the *right information in the right way at the right time* without being asked.

## ABSTRACT

Voice assistants have recently achieved remarkable commercial success. However, the current generation of these devices is typically capable of only reactive interactions. In other words, interactions have to be initiated by the user, which somewhat limits their usability and user experience. We propose, that the next generation of such devices should be able to *proactively* provide the *right* information in the *right* way at the *right* time, without being prompted by the user. However, achieving this is not straightforward, since there is the danger it could interrupt what the user is doing too much, resulting in it being distracting or even annoying. Furthermore, it could unwittingly, reveal sensitive/private information to third parties.

In this report, we discuss the challenges of developing proactively initiated interactions, and suggest a framework for when it is appropriate for the device to intervene. To validate our design assumptions, we describe firstly, how we built a functioning prototype and secondly, a user study that was conducted to assess users' reactions and reflections when in the presence of a proactive voice assistant. This pre-print summarises the state, ideas and progress towards a proactive device as of autumn 2018.

## KEYWORDS

Voice assistants; proactive devices; personalisation; spatial ai; scene understanding; decision making

## 1 INTRODUCTION

In 2014, Amazon announced its Echo speaker with Alexa voice-controlled personal digital assistant. Three years later, the smart speakers represented perhaps the fastest growing market in home appliances with more than 30M devices shipped world-wide [51]. This trend has been rapidly increasing with recent integrations of smart assistants into various HiFi audio systems [45], smart TVs [3] and cars [7]. But what is a voice assistant? What does it currently do? And what else can it *potentially* do? The current generation of voice assistants already performs exceptionally well with basic interactions such as answering knowledge questions (e.g. “What time is it?” “Who is the president of the US?”), integrating 3rd party content providers (e.g. Spotify, Netflix, ...) or controlling various Internet-of-Things (IoT) devices. More advanced products are capable of personalised interactions (e.g. “What is in ‘my’ calendar?” “Do ‘I’ have any unread emails?”) or even short (domain constrained) dialogues [15].

However, the current generation of voice assistants is also somewhat limited in the sense that they are *reactive*, i.e. they “only” *respond* to commands. Moreover, all interactions are initiated by the user using the “voice trigger” keyword. Typically, they do not understand their surrounding environments well; they do not understand where they are, what else in the room is, how many people are around or how they interact with each other. Hence, such devices i) may fail in some situations due to the lack of or misinterpreted context (e.g. Alexa incident [2]) and ii) it is difficult for them to *initiate* non-distracting conversation which significantly limits their capabilities and potential interactions.

\*Equal contribution. Copyright 2020 by the authors.

While considerable amount of effort has focused on extending short user-initiated interactions into longer (20 mins or so) multi-domain dialogues [36], we argue, that if voice assistants are to become smarter, they will need to know how to engage in a conversation, and in particular, when to take the initiative to speak. To do so, devices need to be able to proactively assist the users with a range of activities, reminders and day-to-day routines by learning their habits.

How do designers decide when the device should intervene and what information to use? And how to personalise it for a given person? To begin, we propose that the device should be able to let the user know about important events for the user and more generally in the world. This can be determined and extracted from the user's email accounts/calendar, habits and past digital behaviors. For example the arrival of new email and breaking news about a topic they have shown interest in. It should be announced when it is convenient for the user to attend by assessing their current situation. This could be based on scanning the room for whether the person is alone or with presence of others, what time of day it is, what the person is currently doing and how urgent the information is. At the same time, they should not overload the users with too many verbal updates or disturb them when they are engaged in another task such as having a conversation with someone else. But how to achieve this so that users find it useful and are comfortable with being interacted with in this manner, while not being annoying or finding it too disruptive or intrusive, is an open question. It is difficult to achieve the right balance due to the inherent ambiguity involved, an expected engagement of user attention and because the consequence of an unwanted distraction is significantly more disturbing and irritating when compared to push-like smartphone notifications, which are less invasive. At the same time, collecting the right kind of data that is not considered invasive of someone's privacy is challenging.

One way to tackle this issue is to analyse the (social) context, using external sensory data collection, that detects certain user "states" such as presence by self, level of busyness, emotional reactions and so on. This approach, however, typically does not scale to data amounts and the diversity required by modern deep reinforcement learning approaches mapping raw audio-visual data directly to decisions [46] (the reward signal is also too sparse and indirect as we only have an access to weak and noisy proxy such as user emotional states, and potentially very long spans between causes and effects), however, provides sufficient data for designing and studying proactively initiated interactions. An alternative approach is to detect and determine other aspects of the user's context, and their readiness and willingness to be "spoken" to by a voice-assisted device in their home.

In this paper, we describe how we have designed a proactive robot-based voice prototype with the goal of providing the *right information*, in the *right way* at the *right time*, without being prompted by the user. Our approach is to scan the situation using a form of spatial AI and to limit the kinds of proactive interactions to practical day-to-day tasks (e.g. weather/traffic/press updates, email/calendar notifications). Our focus is on how to use aspects of the context in relation to a user's privacy. Our approach relies on: (i) semantic scene understanding using spatial AI and multi-modal sensory inputs, (ii) semantic content understanding through prioritising types of interactions, iii) fault-tolerant design of the user experience (UX) and (iv) the design of hardware to draw the user's attention from what they are doing.

In the first part of the paper, we describe how we designed our voice assistant prototype to be proactive. It was built using currently available robotic and machine learning technology. To detect context in real time it has been implemented using a novel hardware and software platform equipped with multi-modal sensors. To alert the user to when it is about to speak, the device is programmed to move and light up with a patterned colour display on its body. Then we describe the key elements behind Spatial AI - a method of aggregating relevant statistics across modalities, surrounding space and time. In the remainder of the paper we describe the user study we conducted to evaluate how acceptable, annoying and informative the device was for various conditions, using a living lab experiment.

## 2 RELATED WORK

The current generation of personal voice assistants is *reactive* in the sense that they "only" respond the requests and hence all interactions have to be initiated by the user. A typical interaction with a reactive device [40] proceeds as follows: i) in its idle mode, the device is silently "waiting" and continuously running a small on-device module whose only purpose is to recognise the "voice trigger" keyword (e.g. "Alexa!", "Ok Google!", ...); ii) when such a keyword is detected, user provides her request which is streamed to the cloud where this audio input is processed (speech recognition → natural language understanding → response generation); and iii) the device replies to the user or executes some other interaction (e.g. playing a song, setting a timer, ...).

This process is typically repeated from scratch for any other interaction and often even for a simple follow up. More advanced devices are capable to carry the context over a few more exchanges with the user or to offer a "one more thing"<sup>1</sup> at the end of an interaction, but not to initiate the interaction

---

<sup>1</sup>Some devices provide "one more thing" during or at the end of an interaction initiated by the user (e.g. offering a traffic update after being asked for directions), however, this is different from *proactively initiated interaction*.

itself. What if the smart voice assistants could initiate an interaction or conversation? When and where would it know how to start?

### Initiating proactivity

Proactivity for human-robot interactions requires an estimation of *public* and *social* distances [17] by the robot, so the user is aware of the fact the device exists, is trying to initiate an interaction and the topic of this interaction is also somewhat expected. Various studies [10, 23, 41, 49] that have explored how to approach people (for the first time) in the most effective way found that the user's awareness and understanding of the robot's capabilities is crucial to successfully execute pro-activity. Alerting people to a new digital event (e.g. new text message arrived, breaking news) has been a relatively simple design problem for *personal* digital assistants embedded on smartphones/laptops/tablets as the public and social distances are better defined from the beginning (physical environment is confined to a mutually known environment, and user understands what to expect from the device). The universal uptake of push notifications on smartphones in the last 10 years has transformed how users are updated of new content; not just new emails or text, but also likes, new posts and new pictures uploaded. Smith *et al.* [44] discusses how mobile notification systems affect users and what makes them distracting, and Weber *et al.* [52] how to design them such they are less disruptive to the end user. Users can also be in control of how they manage them - choosing to glance, ignore or open the alerting app, and if the continuous stream of notifications becomes too overwhelming, users can switch them off.

Initiating an interaction from a speech-based robot, however, is quite different. It requires getting someone's attention and knowing if they are receptive to being interrupted. This involves determining when the timing is appropriate while also knowing how to best deliver the content verbally (taking into account user engagement at the moment, privacy, efficiency and other contextual information). If the robot butts in at the wrong moment (e.g. during an intimate moment) or too often it can be annoying and distracting – to the point they will abandon using it. However, distraction is not simply a binary matter; their threshold levels greatly depend on timing (or preceding and current user activity), and that tolerance for their frequency varies between users. This suggests the importance of understanding the contextual setting of the surrounding environment, the value of using personalisation and enabling user adaptation [32, 44]. Weber *et al.* [52] proposed system of aggregation and distribution of notifications between multiple smart devices (primarily based on the user vicinity to one of them), however the aspects related to '*when*' and '*how*' to notify the user, including corresponding privacy issues in multi-user

environments, become paramount. Finding the acceptance threshold may vary from type of notification to type of user.

Another factor that will become more central in considering proactivity is how the robot is perceived in terms of its personality, social and emotional intelligence [8, 9]. Some people may look forward to their friendly chatty robot telling them things – akin to having a friendly person at home who is always chatting. Their ability to switch between being proactive and responsive with people needs to be designed to be natural, acceptable and enjoyable.

The user can choose whether to act upon or ignore a notification appearing on their smartphone or other display. In contrast, voice assistants need to decide when is a good time to notify the user, how many, in what form and in what sequence to present them. One approach to deciding when is an opportune or good time for a virtual assistant to interrupt a user is to analyse conversations in the background [31] assuming there is more than one person in the room having a conversation and that the ambient noise (e.g. cooking, TV on) is not too great. If possible this kind of speech recognition could be used to predict when a user might want to run a search on their phone from their conversation, and would require that the speech system is able to detect topical resources from conversation, and be able to perform a level of semantic analysis. There are also a number of ethical and privacy concerns with using always on streaming as input for proactive interactions.

The amount of updates a voice assistant might conceivably be in control of is likely to be smaller - at least to begin with - when compared with the number of smartphone push notifications typically received - although this could increase as advertisers and app developers discover ways of attracting 'ears'. The real danger - which is not the case for smartphones - is the potential to be more disruptive. It only takes one wrongly timed verbal notification to make someone angry. Another challenge is getting the user's attention - especially when they are attending to something else. What kind of signal is required to get someone to listen to the device?

Few devices to date have been designed to proactively initiated interactions, with some exceptions being social robots like Jibo [22] and Kuri [26] where the presence of the user face (or voice) may trigger some activity conditioned on some auxiliary contextual information (e.g. proactive greeting in the morning, invite to play a short game, telling a joke, etc.)<sup>2</sup>. Voice assistants such as Alexa [1] and Google Home [16], can offer access to personal information (email, calendar) using voice-based authentication, however, in a fully reactive manner, or as a follow up of the ongoing interaction [40]. None offer yet a comprehensive range of reactive

<sup>2</sup>Note, this report was written in late 2018, when these products were being actively developed. As of now both were cancelled.

and proactive interactions, where the device decides when, what and which information to provide.

Moving from reactive to proactive devices is challenging as it *fundamentally* changes the whole interaction process, requiring advanced cognitive capabilities of devices and to some extent also novel hardware. Consider e.g. a “new email” proactive reminder to demonstrate the major challenges and key differences from common push-like phone notifications. Smartphones notify the user as soon as the email is received, either by sound, vibration or simply by silently popping the notification up on the screen in case the user does not want to be disturbed, assuming the user will get back to it *whenever it is convenient*. In some sense, a device can (almost) keep flooding the user with more and more notifications as it is the *user who decides (initiates)* what and when is relevant to herself. This is in sharp contrast to proactive reminders on voice assistants which co-exist with the users in open-world environments (they are not used only when user explicitly controls them). This is a significantly more complex task and the device cannot just “blindly” notify the user as soon as an email is received as the user may not even be around. The device therefore has to first possess some comprehension of the surrounding environment and has to identify whether the user is around. The device has to understand, whether it is convenient to notify the user now as it should not disturb or overload her with too many interactions when she is cognitively engaged (*i.e.* having a conversations or focusing her attention on some other tasks).

### Privacy concerns

If the device concludes that the user should be notified now, it needs to attract the user’s attention before it attempts to deliver the message to give her some time to get prepared and focused. This step is quite different from smartphones, where a subtle buzz or a beep are used. Furthermore, the message needs to be delivered in an appropriate way, based on who is around, e.g. some messages may be private or not be appropriate for kids, and therefore should be delivered when the user is alone. For instance, the device should not ask the user whether she is around or who she is as this could quickly become annoying. Instead, it should do this cognitive process in the “background” and infer it automatically. Proactive interactions need to be designed in a fault-tolerant manner, taking into account potential AI imperfections. This suggests that they need to err on being conservative, initiating interaction, only when confident the user is willing and ready to listen.

What happens when they say something wrong? Should they express human-level responses [18] and apologise even (*cf.* Reeves and Nass [38])? As they become more proactive, would it be desirable for them to look less like inanimate

objects (*e.g.* stationary cylinders) and instead look, animate and behave much more like *robots*?

### Understanding the local context

One approach to deciding when is a good time for a robot or smart speaker to alert a user to a new message is to use cues from the local context. Semantic maps have long been considered to be a prerequisite for decision making systems operating in partially observable 3D environments. This problem is known in robotics as Simultaneous Localisation and Mapping (SLAM) [12], while in biological systems as cognitive maps of the environment [13, 14, 27]. During the past few years, real-time (dense) semantic SLAM has made a significant progress, for instance [19, 30, 39, 50] showed how to build such maps in real-time using only passive cameras or even learn how to segment previously unseen objects on-the-fly [33, 48]. Bhatti *et al.* [6] has also shown recently how semantic maps can be used for learning decision making policies for agents operating in dynamic environments.

Our research is concerned with what new features and underlying model are needed to enable voice assistants to become smarter by taking the initiative to speak up while avoiding situations where they are perceived to be annoying.

## 3 METHODOLOGY

To build the next generation of voice assistants that have the capability of being both proactive and reactive, our research focuses on the following aspects:

- (1) context awareness using spatial AI
- (2) semantic content modelling
- (3) cueing the user’s attention

### Context awareness using spatial AI

Spatial AI is a broad term that refers to building representations for decision making of agents operating in spatial domains. As such, in this work it spans scene understanding, speaker and audio event recognition, spoken language processing (including emotion modelling where necessary) and decision making [12]. We adopt a multi-modal version of a spatial AI; in-built cameras and a microphone array with computer vision and audio processing algorithms are used to infer a richer picture of what is happening.

Our approach is to use a combination of multi-modal semantic scene understanding and decision making subsystems to lay the foundation for proactively initiated interactions. Semantic scene understanding accumulates information from multi-modal sensors that provides a single, unified and machine-interpretable overview of the robot’s vicinity. A decision making subsystem combines this semantic information about the vicinity using different proactivity levels

(see below), user profiles, and meta-data about the past interactions. See Sec. 5 for technical details.

### Semantic content modelling

The importance of a new message or notification to a user will vary (*e.g.* email with “Meeting in 10 minutes?” is likely to be more important than a periodical newsletter). The question this raises is how does the system decide which is most important and which can wait? Our approach uses semantic understanding of content of interactions, where messages are hierarchically stacked, ranging from immediate notifications to periodical batch updates. Table 1 shows our hierarchy of levels of proactivity. Note, these are different from levels of autonomy of a virtual personal assistant [40] to independently execute tasks or make decisions on behalf of its owner (*i.e.* automatically decide on things like shopping, booking travel, scheduling meetings, …).

---

**Level 1 (L1)** – Comprises push-like (*e.g.* new email) and proactive routines (*e.g.* morning news overview). User recognition enables personalised interactions (*e.g.* greetings, personalised reminders) but no requirement for semantic scene or content understanding. All interactions have equal importance, *i.e.* there is no prioritisation. The users is notified of either using daily batch updates (*e.g.* morning/evening routines) or push-like one-off notifications whenever a new email is received. All updates have the same priority.

---

**Level 2 (L2)** – Prioritises messages that are scheduled based on the context. This requires semantic scene and content understanding. A rule that might be used for this is *e.g.* assign higher priorities to breaking news containing keywords (tags) such as “terrorist attack” or “politics”. The use of semantic content understanding enables prioritisation (and “grouping”) of interactions based on their importance.

---

**Level 3 (L3)** – The highest level, where the voice assistant is capable of life-long learning of user habits to keep refining proactive interactions over time. Proactive interactions are embedded into complex dialogues. It would involve the user and system having more in-depth interactions, for example, helping the user to improve on their well-being. The devices should be able to learn automatically user preferences about content or frequency of updates.

**Table 1: Proposed hierarchy of proactive interactions.**

To begin, we focus on levels 1 and 2, in order to determine if there are any differences between these two levels of proactivity on user acceptance and perceived usefulness. As content, we use practical day-to-day updates, as such interactions transferable across different users. The subset of practical day-to-day interactions we implemented for the study were email, calendar, traffic info, news, IoT lights and TV. See Appendix A for some examples of L1/L2 rules.

### Cueing the user’s attention

As part of the move towards creating proactive devices we believe, it is important to consider how to attract the user’s attention to when an update/alert is about to be spoken. We decided to incorporate a form of ambient design into the body of our robot voice assistant, through the use of coloured lights, movement and appropriate synchronisation of the UX such the user has enough time to tune into interaction mode with the device. These choices, including hardware and software considerations, are in detail outlined in the following Section 4.

## 4 SYSTEM OVERVIEW

In this section, we describe our platform from hardware, basic user interaction, and software perspectives.

### Hardware

Our device consists of a fixed base and a moving head (*cf.* Fig. 2 and Fig. 4). It is equipped with two DC motors allowing for continuous 360° rotation around its vertical axis and up to 80° rotation in direction perpendicular to the vertical and horizontal axes. The front side of the moving head consists of a custom circular-shaped LED matrix with 480 RGB LEDs and covers three speakers. Combination of these components enables the device to attract user’s attention, communicate with the user and express various “emotions” or mimic persona types [53] (also see Fig. 4 for a visual example).

The moving head contains two 8 mega-pixel RGB wide-angle cameras rotated by 90° w.r.t. each other (*cf.* Fig. 3) to enable perceiving the surrounding environment under all possible rotations, custom far-field microphone array with 4 microphones and 6-axis inertial measurement unit (IMU). The richer sensory inputs (*i.e.* microphones *and* cameras) allow us not only to process standard audio modality but also combine it with visual data, which significantly extends perception abilities to understand the environment and interactions among the users. Thus, it helps to overcome sensor limitations, *e.g.* 360° sensing of microphone arrays overcomes limited field-of-view of cameras; visual data may help with source disambiguation in noisy areas, multi-user interactions, *etc.* The IMU is used by motors feedback controllers. Device uses 6-core ARM CPU, dual-core GPU with 4 GB DDR4 RAM, 8 GB NAND flash memory storage and is equipped with WiFi and Bluetooth modules and runs an embedded Linux OS.

### User Interface, Interactions and 3rd Party Services

*Basic interactions.* Our device has two basic states, called “idle” and “up” (*cf.* Fig. 2). For the majority of the time, the device is in the “idle” state waiting for the voice-trigger, which is commonly used in all reactive scenarios. However, even in this state, the device detects acoustic events around the



Figure 2: Our platform in its “idle” (left) and “up” (right) states (cf. Sec. 4) with other animations shown in between.

device and can rotate around its vertical axis to “scan” the 360° environment using cameras. Scanning is triggered either periodically or using an arbitrary acoustic event (*i.e.* not a hot-keyword; rather sounds corresponding to events such as walking, doors activity, ...). The second basic state, “*up*”, is primarily used when the user interacts with the device or when the device wants to attract user’s attention to proactively initiate an interaction with her. Our platform supports various transitions between the two and all such animated motions can be combined in arbitrary ways (cf. Fig. 2).

In a standard reactive scenario, the device is in its “*idle*” mode listening for a voice trigger. Once this is provided by the user, the device “*wakes up*” and rotates so that it faces the user (using *Spatial AI* described below) to establish “*eye contact*” with the user. Next, the device is ready to process user’s request as any other voice assistant would. However, it is also able to express emotions using a combination of animated movement and LED matrix (cf. Fig. 4), *e.g.* when the device does not understand the user’s request. Such basic movements and interactions create so called “*presets*” that can be arbitrarily combined to create more advanced (non-verbal) interaction capabilities.

*Unboxing scenario.* To support personalised interactions (potentially with multiple users), our device has to learn how to recognise the user(s) first. This happens during the so-called “*unboxing*” or “*learn me*” interaction, which can be triggered by the user. The device instructs the user to move to various locations in the room to collect multiple views of her face that are used to extract 128 dimensional facial embeddings trained using the triplet loss [35], and similar 128 dimensional speaker embeddings obtained from neural network trained using a teacher-student approach [34].

*3rd party services.* Our device supports numerous services for non-personalised interactions such as weather forecast, headlines and traffic updates as well as personalised interactions such as calendar or email. Our platform also integrates smart TVs, lights, Nest or Sonos as examples of IoT devices.

## Software

Our software pipeline consists of several subsystems (Fig. 5). First, we capture multi-modal data using camera, microphones, IMU and encoders (Fig. 5 A). Audio-visual data are

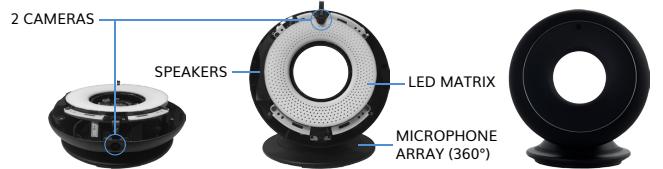


Figure 3: Our platform is equipped with two cameras, microphone array, three speakers, LED matrix and two motors.

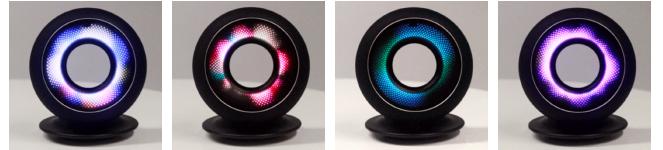


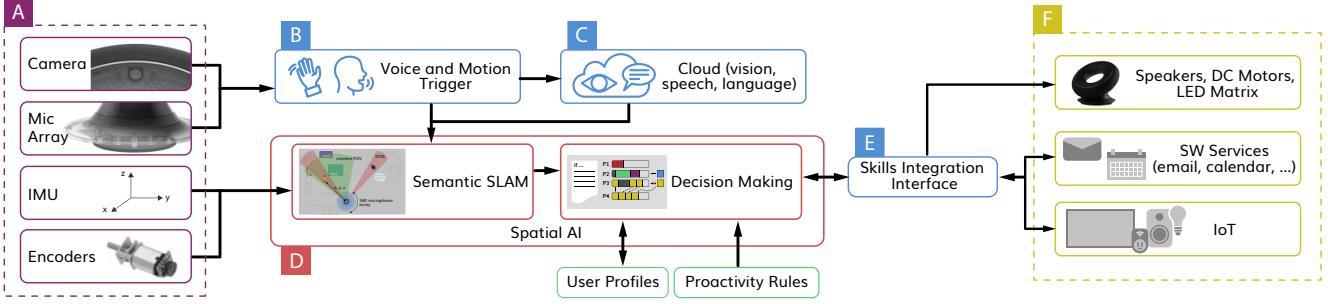
Figure 4: LED matrix is able to show arbitrary animations to attract user’s attention and express emotions.

passed to on-device continuously running voice and motion trigger, whose only purpose is to recognise the hot keyword or detect the motion (Fig. 5 B). When presence of a user is detected and the device has some update ready, or a voice-trigger is spotted, the device wakes up and faces the user [47], starting to process at the same time the audio-visual data. This, depending on the compute requirements can happen either on device or in the cloud (Fig. 5 C) to run more computationally expensive models for computer vision, speech recognition and natural language processing (*e.g.* to map user query to actionable outcome [29]). When necessary, some additional attributes like user’s emotional state [4] or acoustic events [42] may be also estimated. This output is then sent back to the device and combined with data from IMU and encoders in the *Spatial AI* module (Fig. 5 D), which builds a *semantic map* of the environment and is responsible for all *decision making*. This subsystem is supported by a database of user profiles (user preferences, history of past interactions, *etc.*) and proactivity rules. The *Spatial AI* block is connected with a Skills integration interface (Fig. 5 E), which executes actions (play sound, rotate robot, control LED), two-way communication with software services (email, calendar, ...) and IoT devices (Fig. 5 F).

## 5 SPATIAL AI

At the heart of our device lies the *Spatial AI* module [12], a combination of multi-modal semantic scene understanding

## Building Proactive Voice Assistants: When and How (not) to Interact



**Figure 5: Overview of our software pipeline (refer to text for details).**

and decision making subsystems which lays the foundation for proactively initiated interactions:

1. Semantic scene understanding accumulates information from multi-modal sensors available on our platform and provides a single, unified and machine-interpretable overview of the robot’s vicinity. Note, that having a representation capable of accumulating statistics across time is also beneficial to enable lifelong (incremental) learning of users’ habits, however, this is beyond the scope of the paper (L3 devices).

2. Decision making subsystem combines semantic information about vicinity with proactivity rules, user profiles, meta-data about the past interactions, and is responsible for prioritising and scheduling interactions.

### Semantic Scene Understanding

We draw inspiration from [6] and opt for “top-down” views (re-projections) of semantic maps (*cf.* Fig. 6). To this end, we estimate localization of the robot with respect to the environment and in parallel detect important stationary (*e.g.* TV, sofa, ...) and dynamically moving objects (users). In order to update a semantic map from robot’s first-person view at each frame, we accumulate such semantic information by projecting it onto a common 2D map, essentially a “floor-plan” with encoded positions of the robot and objects.

*Model.* We use a multi-modal tracking-by-detection paradigm with probabilistic data association formulated as a Markov Random Field (MRF) [25, 54]. Let  $\mathcal{X} = \{\mathbf{x}_i\}$  denote a set of observations corresponding to detection responses  $\mathbf{x}_i = \{x_i, t_i, a_i, d_i, l_i\}$  where  $x_i$  is the position,  $t_i$  the time step,  $a_i$  appearance and audio features,  $d_i$  the detection score and  $l_i$  the semantic label. A trajectory  $T_k$  is defined as an ordered sequence of observations  $T_k = \{\mathbf{x}_{k1}, \mathbf{x}_{k2}, \dots, \mathbf{x}_{kl}\}$ , where  $\mathbf{x}_{ki} \in \mathcal{X}$ . The goal of the global data association is to maximize the posterior probability of trajectories  $\mathcal{T} = \{T_k\}$  given the set of observations  $\mathcal{X}$

$$p(\mathcal{T}|\mathcal{X}) = p(\mathcal{T}) \prod_i p(\mathbf{x}_i|\mathcal{T}) \quad (1)$$

The likelihood function  $p(\mathbf{x}_i|\mathcal{T})$  of the observation  $\mathbf{x}_i$  is defined by Bernoulli distribution which models the cases of

being a true detection as well as false alarm

$$p(\mathbf{x}_i|\mathcal{T}) \approx \begin{cases} \beta_i, & \text{if } \exists T_k \in \mathcal{T} \wedge i \in T_k \\ 1 - \beta_i, & \text{otherwise} \end{cases} \quad (2)$$

The prior over trajectories decomposes into the product of unary and pairwise terms

$$p(\mathcal{T}) \approx \prod_{T \in \mathcal{T}} \psi(T) \prod_{T, T' \in \mathcal{T}} [T \cap T' = \emptyset] \quad (3)$$

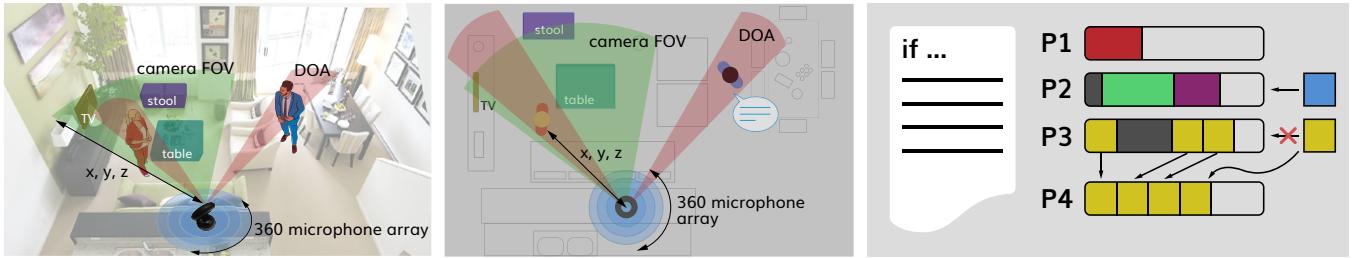
where the pairwise term ensures the trajectories are disjoint. The unary term is given by

$$\psi(\mathcal{T}) = \psi_{en}(\mathbf{x}_{k1}) \psi_{ex}(\mathbf{x}_{kl}) \prod_{i=1}^{l-1} \psi_{li}(\mathbf{x}_{ki}, \mathbf{x}_{k_{i+1}}) \quad (4)$$

where  $\psi_{en}$ ,  $\psi_{ex}$  and  $\psi_{li}$  encode likelihood of entering a trajectory, exiting a trajectory and linking temporally adjacent observations within a trajectory. Note, that our representation could also naturally accommodate dense(r) representations (semantic segmentation, material prediction, ...) and dense 3D reconstruction if needed, as it has been shown in [6].

*Inference.* Taking a negative logarithm of (1) turns the maximization into an equivalent energy minimization problem which can be mapped into a min-cost flow network and efficiently solved using an online min-cost solver with bounded memory and computation [28, 54]. We periodically re-run this inference step in an asynchronous thread.

*Appearance features.* We use similar association features to Lenz *et al.* [28], *i.e.* an LAB colour histogram, patch similarity, bounding-box overlap, bounding-box size, location and class label similarity (*cf.* [28] supp.). In order to detect the bounding boxes, we exploit prior knowledge about the scene. The stationary objects (TV, sofa, ...) are detected using the YOLO object detector [37] running as an asynchronous service in a cloud. The predicted bounding-boxes are directly fed into the MRF. However, this would result in too large latency for user detection, tracking and recognition (they are not stationary). Therefore, we run a second, lightweight, dlib frontal face detector [24] on a device GPU which (re)-initializes the fast DSST trackers [5, 11] running in asynchronous threads



**Figure 6: Spatial AI.** Semantic scene understanding accumulates information from multi-modal sensors (left) and provide a single, unified and machine-interpretable overview of the robot’s vicinity (middle) which is used by decision making (right).

to achieve interactive framerates. Such outputs are used directly (e.g. to maintain an “eye contact” with the user within the camera frustum) and as inputs into the MRF. Note, that we could have used a single model suitable for embedded devices such as MobileNet [21], however, this is rather an implementation detail beyond the scope of the paper.

**Audio features.** For acoustic event detection, we use log mel filterbank features extracted from a raw audio signal followed by a convolutional neural network producing per-class posterior probabilities [20, 42]. To take into account co-occurring audio events, we notify the spatial model about each acoustic event that surpasses the expected threshold. Additionally, we estimate direction of arrival (DOA)  $\theta_s$  for each of the detected sounds ‘ $s$ ’ using a set of DOA estimates from the raw signal (as many as detected acoustic events at each given time step), which are then mapped to  $x, y, z$  coordinates<sup>3</sup>. This process can leverage an additional semantic information from vision stream, as shown in [47]. The most likely pairs {acoustic\_event,  $\theta_s$ } for co-occurring events are estimated in the spatial model using visual data.

**User recognition for personalised interactions.** Whenever we detect a face or a spoken acoustic event, we extract an embedding vector and associate it with a particular object trajectory  $T_k$ . At each observation, the embedding vector is classified as a known or unknown user (*i.e.* open-set recognition) using standard feature thresholding and discriminating w.r.t. other known users and mean (background) models. The confidence scores are accumulated across time to avoid per-frame independent decisions and “flickering” predictions.

## Decision Making

Learning a decision making agent is non-trivial due to the lack of training data and sparsity of the reward signal. Additionally, our primary goal is to validate our design assumptions. Hence, we use a manually designed first-order logic decision rules. This makes the system flexible enough (we

<sup>3</sup>In far-field, one cannot easily estimate the distance between sound source and microphone array, thus we assume constant radius when mapping from polar to Cartesian coordinates.

can quickly modify interactions) and at the same time remains easily interpretable (easy to understand failures).

**Interactive rules.** Let  $\mathcal{R} = \{r_i\}$  be a set of interactive rules defined by tuples  $r_i = \{p_i, e_i, t_i^s, t_i^c, o_i\}$ , where  $p_i$  is the priority,  $e_i$  the time span since previous interaction,  $t_i^s$  a triggering service (e.g. received email),  $t_i^c$  is the triggering configuration (e.g. interact if user is the only person around) and  $o_i$  is the set of output actions (LED, speaker, …). Note that multiple rules can be combined together by using them as triggering service  $t_i^s$  (e.g. “weather update” can be appended to “calendar reminder”). We run an asynchronous thread periodically checking all active rules and their associated trigger events. Note that multiple interactions might be triggered at the same time or before the current one finishes. Hence, all triggered interactions are pushed into the scheduler to ensure the user is not overloaded. This does not prevent reactive interactions initiated by the user using a voice trigger; such interactions are simply pushed into the scheduler with the highest priority reserved for the reactive mode, *i.e.* immediate responses to requests initiated by the keyword phrase.

**Scheduling.** We use a multilevel feedback queue scheduler [43], which groups interactions into  $P$  queues. We use linked lists implementation to support iterating over jobs and job removal from the middle of the queue. Each queue is assigned a priority and has its own scheduling algorithm; we use first-in-first-out scheduling. This ensures that an interaction is executed when all the queues with higher priority have been completed. In contrast to a multilevel queue, jobs can move between the queues which prevents starvation of lower priority tasks, and “jobs recombination” can transform multiple tasks in the same queue into a batch (e.g.  $N$  email newsletter updates → single update “the user has  $N$  emails”) and either push it back to the same queue, promote to a higher or demote to a lower priority queue (*cf.* Fig. 6, right).

**Accumulating meta-data.** For each trajectory  $T_k$ , we maintain a fixed size queue of last  $N$  interactions, time and priority of their execution. Such statistics are essential for multi-level scheduling described above. Thus we propagate it to the user profiles, from which, it can be retrieved to help with situations when a user *e.g.* leaves the room for a few minutes.

### Computational Efficiency and Scalability

In addition to on-device processing resources, the current spoken language understanding stack (speech recognition + natural language understanding) runs in the cloud with final latency of around 200 – 300ms as measured from the point when the user has finished her query.

Part of the computer vision stack (object detection, facial embedding extraction) which runs in cloud uses Nvidia Titan X and average processing time takes around 150 – 200 ms. However, it should be noted this latency influences only the first “user detection”, as then we interleave detection with fast on-device object trackers running at 20fps.

Clearly, the amount of high-end hardware required to run our prototype is relatively high, however, it i) is possible to replace many computationally expensive parts of our pipeline by their lightweight alternatives suitable for embedded devices such as MobileNet [21]; ii) not every single device needs to have dedicated hardware, it should rather be shared by multiple devices which could efficiently use minibatching to optimize the cost.

## 6 USER STUDY

We conducted a user study in a living lab, set up to determine how people would react to our proactive robot that is implemented using our spatial AI model. The study was designed primarily to investigate how varying the amount and type of digital content with respect to L1 and L2 interactions impacted on how users found the different kinds of updates to be useful, distracting or even annoying. We also investigated the extent to which different contexts affected the user’s perceptions and what was considered an acceptable update frequency in differing contexts (by varying the scenario and frequency of updates) and what was the effect of being alone or in the presence of someone else. Another variable we were interested in was how to get someone’s attention when they are involved in another task. Would a dynamic cue manifested in the device’s head orienting towards the user and its LED pattern appearing be able to draw their attention without it being annoying? To measure people’s perceptions and reactions, we observed them during the study when subjected to different kinds of proactive interactions initiated by the device and then interviewed them afterwards. Two scenarios were set up: one during a routine period of the day and the other during a non-routine lazy part of the day. This enabled us to explore whether level of busyness affected acceptance and perceived usefulness of the proactive interruptions.

**L1 versus L2 levels.** The content proactively spoken by the device was for practical day-to-day routines. These were: email, calendar notifications, headlines, weather and traffic

updates. Half the sessions were run as L1 interactions - where the device is able to recognise the user, but where no spatial modeling is taking place. The updates are one-off. The other half were run as L2 interactions, where the system uses spatial modelling to detect where the participants are and what they were doing. This provided us with a base to decide when to pro-actively intervene.

For either mode, we used deterministically pre-defined digital content (user’s mailbox, calendar, news ...) to ensure reproducibility for all participants. The news content was synthesised by creating fake news (e.g. *Donald Trump has resigned*). The relative importance was determined by its assumed level of interest. For example, the news headline *Donald Trump has resigned* was assigned to be more important than *Tesco starts selling cars*.

For both L1 and L2 levels, the user was presented with the same number of messages in a session. For L1, however, their arrival was random, and our device notified the user as soon as they were received. L2, on the other side, prioritised and batched the arriving messages. Batching was done based on importance, but also privacy considerations (*i.e.* personal message even if important, should not be read in case user is not alone).

The context was varied for the study. **Condition 1** was designed to simulate relatively short repetitive parts of a day, (e.g. morning routines). **Condition 2** was designed to simulate longer interactions (e.g. lazy afternoon or weekends). As such Condition 1 was designed to take 20 minutes, while Condition 2 was longer and took 40 minutes. The number and types of messages announced in each mode is reported in Table 2. Example email messages for the case where the device was expected to preserve privacy are shown in the Appendix B.

	Email	Calendar	Other updates
Condition 1	6	4	2
Condition 2	16	6	4

**Table 2: Types and the number of messages communicated to participants in each condition.**

**Static versus animated device.** To test the importance of attracting people’s attention before speaking an update, the device was programmed to work in two modes (i) scanning the room in its “idle” mode, and (ii) using an animated motion with lights appearing on the display. This process is shown in Fig. 2, where on the left device is in its sleeping “idle” state, smoothly transitioning to the “wake” interaction mode on the right. The duration of this transition was configurable, but in our experiments took around 2 seconds.

**Alone or in presence of other person.** The study was designed so that for half the time participants sat with someone else and the other by themselves. Participants did not know each other prior to the session. The device operated either in L1 or L2 modes (for the whole session).

### Participants and Protocol

The study took place in London, UK. 20 participants (10 females and 10 males), aged between 18 and 36, were recruited and asked to come three times over a period of two weeks on different days to the living lab. Each participant was paid 15 pounds per hour.

Two participants, who did not know each other prior to the meeting, were brought into the lab that was furnished like a living room. The device was set up in the corner of the room, while the participants were sitting on a couch. The rest of the room was furnished in a usual way, with chairs, coffee table, shelves, wardrobe and TV.

The participants were familiarised with the experiment and device with an instruction brief (attached in Appendix C). The instruction shortly stated the purpose of the research, the plan for the following two sessions as well as the top level summary of what the device could and could not do in the study. Participants were also instructed about the quizzes, cognitive tasks, an exit questionnaire and the importance in participating in the other two sessions in different days. They were then asked if they had any follow up questions, which were then answered verbally.

The first participant was asked to engage in a primary task - which depending on the session was either watching a short movie, or reading a story. Apart from this cognitively engaging task, participants were encouraged to interact with each other as they would normally do if they were sharing home environment.

For the condition 1, the participants were asked to imagine they had been away for a while (*i.e.* morning after waking up), so the device has accumulated lot of content for them, including new emails, remainder and press news. Condition 1 tested how the device should approach the notification process, *i.e.* in the order messages arrived or in a batched manner taking into account priorities of each notification, as well as privacy concerns. In L2 the participant was distracted sporadically, using contextual cues to try to prevent them from becoming overloaded. This session lasted about 20 minutes. After the first session, the participants were given a short break of 15 minutes.

The second session took 40 minutes and focused on Condition 2 content. For this session the participants were told to imagine that they are enjoying a lazy weekend afternoon in order to better understand how their preferences about frequency and types of updates changes with the amount of time spend with the device. The other difference was the

device did not have anything upfront to announce, but rather tried to keep up with the arriving content. In L1 mode, these were being announced immediately as they arrived while in L2 the device tried to prioritise, batch and preserve privacy.

The rationale for having these two different scenarios was to investigate degree of distraction someone is happy to accept under various situations (idle, chatting with others, cognitively engaged on some task) and how important it is to attract user's attention prior to the interaction.

The idle and conversation aspects were embedded by design (users had some time alone with a device, as well as were encouraged to speak with each other). For the part where participants should focus on some cognitively engaging task, they were asked to answer some quiz like questions. For condition 1 these were short movie clips (*i.e.* 5 minutes long) and the corresponding quiz participants were asked to solve (based on the content of the video). Condition 2 involved a 10 minute long reading exercise and the related answer quiz. These were to assess to what degree the device's announcements affect the participants' ability to focus under different operation modes (L1/L2). As shown in the instruction form (see Appendix C), the participants were instructed to give answers only when they managed to learn the answer (no need for guessing).

The order of conditions were counterbalanced as well as the order of situations using the Latin square design. During each session, two researchers observed the two participants' behavior and their interaction patterns with the device. Half way through each session one participant was asked to leave the room. The reason for this is that we also wanted to test how participants would react when just by themselves. Would they feel more comfortable? How would being by oneself differ from being with someone else when the robot proactively spoke to them? How would their approach change when device shared private messages?

At the end of the 2nd session in each day, the participants filled in the survey (see Appendix C) that asked them about their experience of a device notification system, and several short questions about the content of notifications comparing their experiences of whether L1 or L2 approaches were perceived to be more accurate in delivering content.

### Findings

The observations made during the sessions and the answers from the survey revealed that overall most participants appreciated the benefits of having proactively initiated interactions. In general, they liked the appearance of the device, and were not too concerned that it needed to scan the room in idle state. P4 commented, "Enjoyed the proactivity of the robot, no need to check your phone for updates", P6: noted "concept of the machine knowing what updates to give and when". The least impressed participant (P15) agreed that

"proactive updates are useful in terms of interaction". Participants also commented on liking the variety of content (about meetings, calendar events, traffic and weather). Some said they would have appreciated having some control over pausing or stopping them. For example, P15: said there should be the option to "pause updates if there are too many... or simply opt-out". P5 wanted to be able to "ask whether the user wants details or simply skip to speed up going through too many accumulated news". P6: would have liked to be able to "customize / filter updates" while P2 wanted to be able to "request the device to repeat the last update".

Many of the updates read out what turned out to be long texts (email, news). Participants wanted these to be shorter summaries as it was hard for them to pay attention for these lengths of time. Other participants complained about not being able to control the volume of the updates (P10) and the tempo of updates.

(i) *Differences between L1 and L2.* There were not many differences in the participants' comments for the low-frequency content, only a few participants noticed the ranking was not good enough for L1 settings (*i.e.* lack of priorities in this mode resulted in switching contexts between different types of messages). However, this differed dramatically for high-frequency content, where participants showed a strong preference towards L2. In particular, the participants appreciated the assigned priorities and updates given in batches for the L2 settings, for example, P15 noted: "Updates are good. Importance ranking is good", while P8 and P10 both liked batch updates of news.

This difference was even more pronounced for the L1 setting, which participants found very invasive and disturbing. For example, P14: stressed how they "couldn't focus on my task", P6 said "I found it invasive, couldn't concentrate on the tasks", P13 pointed out "there were too many updates, barely had time to think", P3, P7 and P17 all said that there was a lot of information which was quite distracting.

Similarly, participants did not like the way the updates were just announced with no apparent reason behind them. P11 for example, said they "found it very annoying as it just threw information at me in a random order", while P1, P2, P15 and P16 said they would prefer if the important updates had been read first for batched updates. Others would have liked to have more control over the length of email and news updates that were read out.

Participants in general liked that when the device was set to L2, it postponed all personal updates until they were alone (note, in L1 device did not use spatial AI module, thus did not track if there is more than one person in the room and as such could leak some content of the private correspondence. More on this in the next sections.). Participants also pointed out that updates should not be given when they were talking

with others, which point towards another important aspect of spatial understanding of the environment.

(ii) *Stationary versus animated device.* Most of the participants liked the animated device, for example, P2 liked that the device faced me when giving an update", P11 also liked "the way it moved before giving any updates "P16, thought that it acted eye-contact , giving "a good idea of when the device will speak" and P7 and P9 thought it was good that the "device wakes up before giving an update". Conversely, participants complained when the device provided an update without attracting their attention in this way – having gotten used to it. P11, for example, noted how "I didn't like how it gave no warning that it was about to update us so sometimes I missed the first part of the update", while P3 was annoyed as "I wasn't prepared for the updates", and P9 also said "I didn't like that it did not wake up before giving an update".

(iii) *Naive vs privacy-preserving updates.* Almost all participants immediately found personal updates in the presence of other people annoying, for example P11 said "Found it very annoying as it told me private things when someone else was in the room.". See examples in Table 3 in the Appendix B. On the other hand, participants much preferred the L2 type interactions with personal updates given when they were alone. P7 for example, said "It was great when the device said: 'Looks like we're finally alone, so I can update you about personal matters'". In fact, the importance of privacy was the most widely discussed topic that was raised. One participant (P9) became aware of how it could be problematic, "the door was open when updating me, meaning others could listen in easily". To deal with the privacy concern, several participants suggested that the device should request permission before giving out personal information (P1, P10, P12, P15).

## 7 DISCUSSION AND CONCLUSIONS

In this paper, we have proposed that *proactively initiated interactions* will be one of the defining factors of the next generation of voice assistants, identified design principles for such devices, and since not all interactions are equally complex, defined their classification levels. In addition to that, we have described an end-to-end prototype comprising of novel hardware and software, which we used to conduct a live-lab user study to validate our design assumptions.

One of the key assumptions our user study confirms is, that the privacy truly matters to the users<sup>4</sup>. In fact, it represents the key challenge for *proactively initiated interactions* as majority of *important* updates are typically quite personal. As such, it not only introduces new demands on the hardware and software stack to ensure the updates are provided only

---

<sup>4</sup>The study was conducted in London, UK. This might differ in other parts of the world.

when sensitive information cannot be accidentally leaked to the 3rd parties, but also calls for strong legislation and data protection (most voice assistants are used in home environments). From this perspective, it is highly positive to see efforts of some governments putting such legislation in place (most notably GDPR); and to see that many companies go even beyond and consider data privacy to be a *human right*. We would like to stress, that while such legislation and compliance comes with certain (development) costs, it is absolutely critical; not just for the end users perspective, but also for the actual progress of smart voice assistants as it “protects” the *developers* (suppresses fear that the technology they are contributing could be easily misused).

On a hardware side, our prototype device is equipped with the necessary sensors and features to implement proposed pro-activity levels (both perception- and expression- wise). The proposed Spatial AI model is generic enough for fusing various sources of sensory and high level information required to understand the immediate environment and to accordingly initiate interactions. However, if such device were to be deployed beyond the lab environment one would need to carefully communicate when, how and where the device is using the information it has access to, both in real-time when making ad-hoc decisions (*i.e.* it is clear for the user when cameras/mics are on/off and what information exactly the device is looking for in each stream) but also beyond that (*i.e.* is the information stored for further processing / models re-estimation? If so, where - on the device or in the cloud? Who has access to it? May it be manually inspected or annotated at any point? Can the user conveniently reply and manage stored episodes? *etc.*). Some of these issues are being addressed by legislation (like the aforementioned GDPR) but much remain to the terms and conditions and often implementation details.

In the experiments carried in this work, the participants did not raise device-related privacy concerns, but it does not mean they would not have any if the device was at their place. Likewise, we did not answer if they would be happy with the additional privacy trade offs required to provide good experience of proactive interactions (cameras / mics on) when compared to the current reactive devices (only mics on). In either case one would require configurable mechanisms and related functionality to back off to more limited capabilities if the user wishes to (temporarily) disable any of the modalities.

Finally, we would like to stress that while proactively initiated day-to-day interactions (email, calendar, press, ...) exhibit promising potential and demonstrated benefits to the users, we are only at the very beginning. It might be very tempting to start promising interactions proactively improving users’ well-being, and in general, imitating user’s *best friend* (proactive suggestions to “go to a therapy”, “have

a glass of wine” or “told joke to improve user’s mood”). However, we need to keep in mind that such interactions are much less transferable across different users, often depend on user’s personality, current mood and in general require much better understanding of (cultural, social, ...) context. While AI has been making great progress, with its current state, we are nowhere near devices that could support such interactions. Thus, we need to select interactions which we try to transform into proactively initiated very carefully.

## Acknowledgements

In no particular order, we would also like to thank X. Chen, M. Zhou, A. Ye, J. Zhao, P. Tesh, J. Grant, A. Khan, J-C Passepong, G. Groszko, M. Shewakramani and T. Wierzchowiecki for their contributions during various stages of this project.

## REFERENCES

- [1] Alexa 2018. Amazon Alexa. [https://en.wikipedia.org/wiki/Amazon\\_Alexa](https://en.wikipedia.org/wiki/Amazon_Alexa). Accessed: 2019-09-14.
- [2] Alexa Incident 2018. Amazon explains how Alexa recorded a private conversation and sent it to another user. <https://www.theverge.com/2018/5/24/17391898/amazon-alexa-private-conversation-recording-explanation>.
- [3] Alexa TV 2017. Smart TV with Alexa control. <https://www.techdoze.net/how-to-control-a-smart-tv-with-amazon-alexa/>.
- [4] Rory Beard, Ritwik Das, Raymond WM Ng, PG Keerthana Gopalakrishnan, Luka Eerens, Paweł Swietojanski, and Ondrej Miksik. 2018. Multi-modal sequence fusion via recursive attention for emotion recognition. In *CoNLL*.
- [5] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr. 2016. Staple: Complementary Learners for Real-Time Tracking. In *CVPR*.
- [6] Shehroze Bhatti, Alban Desmaison, Ondrej Miksik, Nanthas Nardelli, N. Siddharth, and Philip H.S. Torr. 2016. Playing Doom with SLAM-Augmented Deep Reinforcement Learning. In *arXiv preprint arXiv:1612.00380*.
- [7] BMW voice assistant 2018. BMW launches a personal voice assistant. <https://techcrunch.com/2018/09/06/bmw-launches-a-personal-voice-assistant-for-its-cars/>.
- [8] Cynthia Breazeal and Lijin Aryananda. 2002. Recognition of affective communicative intent in robot-directed speech. *Autonomous robots* (2002).
- [9] Cynthia Breazeal, Cory D Kidd, Andrea Lockerd Thomaz, Guy Hoffman, and Matt Berlin. 2005. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ international conference on intelligent robots and systems*.
- [10] Elizabeth Broadbent. 2017. Interactions With Robots: The Truths We Reveal About Ourselves. *Annual Review of Psychology* (2017).
- [11] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. 2017. Discriminative Scale Space Tracking. *T-PAMI* (2017).
- [12] Andrew J. Davison. 2018. FutureMapping: The Computational Structure of Spatial AI Systems. *CoRR* abs/1803.11288 (2018). <http://arxiv.org/abs/1803.11288>
- [13] Niv Y. Daw, N. D. and P. Dayan. 2005. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience* (2005).
- [14] R. J. Dolan and P. Dayan. 2013. Goals and habits in the brain. *Neuron* (2013).

## Building Proactive Voice Assistants: When and How (not) to Interact

- [15] Google Duplex 2018. Google Duplex. <https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html>.
- [16] Google Home 2018. Google Home. [https://en.wikipedia.org/wiki/Google\\_Home](https://en.wikipedia.org/wiki/Google_Home). Accessed: 2019-09-14.
- [17] Edward Twitchell Hall. 1966. *The hidden dimension*. Vol. 609. Garden City, NY: Doubleday.
- [18] Adriana Hamacher, Nadia Bianchi-Berthouze, Anthony G. Pipe, and Kerstin Eder. 2016. Believing in BERT: Using expressive communication to enhance trust and counteract operational error in physical Human-robot interaction. *RO-MAN* (2016).
- [19] Alexander Hermans, Georgios Floros, and Bastian Leibe. 2014. Dense 3D semantic mapping of indoor scenes from RGB-D images. *ICRA* (2014).
- [20] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. 2017. CNN Architectures for Large-Scale Audio Classification. In *ICASSP*.
- [21] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR* (2017).
- [22] Jibo 2018. Jibo. <https://www.jibo.com/>. Accessed: 2019-09-14.
- [23] Yusuke Kato, Takayuki Kanda, and Hiroshi Ishiguro. 2015. May I Help You?: Design of Human-like Polite Approaching Behavior. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*.
- [24] Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *JMLR* (2009).
- [25] D. Koller and N. Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.
- [26] Kuri 2018. Kuri. <https://www.heykuri.com/>. Accessed: 2019-09-14.
- [27] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2016. Building Machines That Learn and Think Like People. *CoRR* abs/1604.00289 (2016).
- [28] Philip Lenz, Andreas Geiger, and Raquel Urtasun. 2015. FollowMe: Efficient Online Min-Cost Flow Tracking with Bounded Memory and Computation. In *ICCV*.
- [29] Xingkun Liu, Arash Eshghi, Paweł Swietojanski, and Verena Rieser. 2019. Benchmarking Natural Language Understanding Services for building Conversational Agents. In *10th International Workshop on Spoken Dialogue Systems Technology*.
- [30] John McCormac, Ankur Handa, Andrew J. Davison, and Stefan Leutenegger. 2017. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In *ICRA*.
- [31] Donald McMillan, Antoine Lorette, and Barry Brown. 2015. Repurposing Conversation: Experiments with the Continuous Speech Stream. In *ACM CHI*.
- [32] Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. 2016. PrefMiner: mining user's preferences for intelligent mobile notification management. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM.
- [33] Ondrej Miksik, Vibhav Vineet, Morten Lidegaard, Ram Prasaath, Matthias Nießner, Stuart Golodetz, Stephen L. Hicks, Patrick Perez, Shahram Izadi, and Philip H. S. Torr. 2015. The Semantic Paintbrush: Interactive 3D Mapping and Recognition in Large Outdoor Spaces. In *ACM CHI*.
- [34] Raymond W. Ng, Xuechen Liu, and Paweł Swietojanski. 2018. Teacher-Student Training for Text-independent Speaker Recognition. In *Proceedings of the IEEE Workshop on Spoken Language Technology*.
- [35] O. M. Parkhi, A. Vedaldi, and A. Zisserman. 2015. Deep Face Recognition. In *BMVC*.
- [36] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. 2017. Conversational AI: The Science Behind the Alexa Prize. *CoRR* (2017).
- [37] Joseph Redmon and Ali Farhadi. 2016. YOLO9000: Better, Faster, Stronger. *arXiv preprint arXiv:1612.08242* (2016).
- [38] Byron Reeves and Clifford Ivar Nass. 1996. *The media equation: How people treat computers, television, and new media like real people and places*. Cambridge university press.
- [39] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H. J. Kelly, and Andrew J. Davison. 2013. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *CVPR*.
- [40] Ruhi Sarikaya. 2017. The Technology Behind Personal Digital Assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine* (2017).
- [41] Satoru Satake, Takayuki Kanda, Dylan F Glas, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. 2009. How to approach humans?: strategies for social robots to initiate interaction. In *International conference on Human robot interaction*. ACM.
- [42] Rubio Shi, Raymond WM Ng, and Paweł Swietojanski. 2019. Teacher-student training for acoustic event detection using audioset. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- [43] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. 2008. *Operating System Concepts*. Wiley Publishing.
- [44] Jeremiah Smith, Alessandra Russo, Anna Lavygina, and Naranker Dulay. 2014. When did your smartphone bother you last?. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM.
- [45] Sonos 2017. Sonos One. <https://www.sonos.com/en-gb/alexa-on-sonos>. Accessed: 2019-09-14.
- [46] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement learning - an introduction*. MIT Press.
- [47] Paweł Swietojanski and Ondrej Miksik. 2020. Static Visual Spatial Priors for DoA Estimation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- [48] Julien Valentin, Vibhav Vineet, Ming-Ming Cheng, David Kim, Jamie Shotton, Pushmeet Kohli, Matthias Niessner, Antonio Criminisi, Shahram Izadi, and Philip H. S. Torr. 2015. SemanticPaint: Interactive 3D Labeling and Learning at your Fingertips. *ACM Transactions on Graphics* (2015).
- [49] Dominique Vaufreydaz, Wafa Johal, and Claudine Combe. 2016. Starting engagement detection towards a companion robot using multimodal features. *Robotics and Autonomous Systems* (2016).
- [50] Vibhav Vineet, Ondrej Miksik, Morten Lidegaard, Matthias Nießner, Stuart Golodetz, Victor A. Prisacariu, Olaf Kähler, David W. Murray, Shahram Izadi, Patrick Perez, and Philip H. S. Torr. 2015. Incremental Dense Semantic Stereo Fusion for Large-Scale Semantic Scene Reconstruction. In *ICRA*.
- [51] VoiceBot.AI 2018. VoiceBot.AI Smart Speaker Market Analysis. <https://voicebot.ai/2018/01/07/56-million-smart-speaker-sales-2018-says-canaly/>.
- [52] Dominik Weber, Alireza Sahami Shirazi, and Niels Henze. 2015. Towards smart notifications using research in the large. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. ACM.
- [53] Steve Whittaker, Yvonne Rogers, Elena Petrovskaya, and Hongbin Zhuang. 2020. Designing Personas for Expressive Robots: Personality in the New Breed of Moving, Speaking and Colourful Social Home Robots. *ACM Transactions on Human Robot Interactions* (2020).
- [54] Li Zhang, Yuan Li, and Ramakant Nevatia. 2008. Global data association for multi-object tracking using network flows. In *CVPR*.

# Appendices

## A EXAMPLES OF DECISION MAKING RULES

### L1 rules:

- Personalised greetings:
 

```
if user_detected_1st_time & user_recognised then
    message ← Hi $User
else if user_detected_1st_time & !user_recognised
then
    message ← Hi
end if
```
- Weather:
 

```
if user_detected_1st_time_a_day then
    update about weather
end if
```
- IoT lights:
 

```
if user_detected & time ≤ 9am then
    turn lights on
end if
```
- Calendar:
 

```
if event ≤ 2 hours then
    remainder_priority ← high
end if
if event_same_day then
    remainder_priority ← medium
else
    remainder_priority ← low
end if
```
- Other (email, news, etc):
 

```
if new_event then
    put into scheduler (push-like)
end if
```

### L2 rules (services):

- Email:
 

```
if whitelisted {family, boss, friends} email then
    importance ← high
else if spam or newsletters then
    importance ← low
end if
```
- News:
 

```
if contains {terrorist, politics} then
    priority ← high
end if
```

### L2 rules (scheduling / meta-rules):

- if personal\_update {email, calendar} then
 

```
postpone until user is alone
end if
```

- if event\_type exists in Queues then
 

```
if event_importance is high then
    combine into a single one-by-one interaction
else
    combine into batched interaction
end if
end if
```
- if news & weather then
 

```
first update news
end if
```
- if news & calendar then
 

```
first update important news then unimportant
calendar
end if
```
- if news & email then
 

```
first update email
end if
```
- if calendar & email then
 

```
first update important calendar
end if
```
- if time\_last\_update - time\_elapsed s < T\_min then
 

```
schedule interaction in (T_min - time_elapsed)
end if
```

Note, that  $T_{min}$  is a constant specific for each interaction type (and priority)

## B EXAMPLES OF PERSONAL MESSAGES

Here we show examples of "personal" notifications, for which device should preserve privacy (read them only when user is alone).

---

You have a new email Lloyds bank. It says: Hi [user], You still have not paid back your debt and you have only 500 pounds at your account. Therefore your credit card will be blocked.

---

You have a new email from Your Boss. It says: Hi [user], I'm very much unhappy with your performance and hence decided to put you on performance improvement plan. You have 6 months to prove your value, otherwise you will be terminated. Let's have a chat about it later today.

---

You have a new email from Thames Water. It says: Hi [user], please give us meter readings by end next week

---

You have a new email from Pedro. It says: Hey [user], pub at 6? [user], I'm supposed to remind you a date with Anna tonight.

---

You have a new email from Jeniffer. It says: Hey [user], dinner at my place at 7?

---

[user], I'm supposed to remind you to fire an intern next week.

---

You have a new email from Jeniffer. It says: Stop stalking me!!!! Next time I'm gonna call police you bastard!!!

**Table 3: Examples of the private messages used during living lab sessions. The [user] tag was used to personalise device interactions with each participant.**

## C PARTICIPANT FORMS

### Entry instructions

**Hi!** Nice to meet you! And thank you very much for participating in this study!

Our study is concerned with how people can benefit from transforming reactive devices, which typically only respond to requests initiated by the users (see blue and orange speech bubbles), to proactive ones offering the user the right information in the right way at the right time without being asked (see green speech bubble).



#### Instructions:

- You will be placed in a room, given a task (reading, watching TV) and a question sheet. Your goal is to answer as many questions as possible but do **NOT** try to guess if time runs out or you are uncertain - we're not grading you :)
- During this task, the device will update you with some information. After each session, you'll be given a questionnaire in which you'll answer: i) some questions about given updates, ii) how did you like the interactions
- In this study, you cannot ask the device any questions
- Content matters (importance of updates, frequency, ...) not style (used "language", personality of the speaker, ...)
- There will always be 2 people in the same room; the first one has to work on the task, while the second one ("unknown") always leaves the room in the middle. Participants are encouraged to communicate with each other, have fun, ...
- During each session, the updates will be given in a different form. On the last day, you'll fill in one more form asking you which session you liked the most and why

#### Let's practice a bit:

1. Read the form with questions
2. Let's now play some sample interactions
3. Now, you'll be working on the task
4. So the sample interactions finished so now you'll be asked to fill in the original form

## Answer sheet

Team ID:

Session ID:

Name 1 (known user):

Name 2 (leaves in the middle):

	Question	Answer
“Quality”	Did the device provide useful updates?	scale: 1 (very useful) - 5 (not useful)
	Were updates given in a logical order?	scale: 1 (very much) - 5 (not at all)
	Would you prefer if more updates were given explicitly one-by-one, or would you prefer more updates “grouped” to batches?	A. more 1-by-1 B. about right C. in batches
	Was there any personal info leaked to someone who should not have heard it?	A. yes B. no
	Were you able to focus on your task?	scale: 1 (very much) - 5 (not at all)
Device behaviour	How important is/would be an eye-contact with you before you get an update?	scale: 1 (very important) - 5 (not important)
	How annoying was the device moving in the background?	scale: 1 (very annoying) - 5 (haven't even noticed it)
	Would it be useful to have a chance to ask the device to repeat the update?	A. not much (easy to understand most of the updates) B. yes (I've misunderstood too many updates)
Scheduling	How would you rate the number of updates?	A. too little B. about right C. too many
	How would you rate frequency of updates?	scale: 1 (very good) - 5 (poor)
	How would you score the timing of the updates?	scale: 1 (best) - 5 (worst)
General	What did you like in general? (short text)	
	What didn't you like in general? (short text)	
	What would you do differently? (short text)	

## **Exit questionnaire**

Team ID:

Session IDs:

Name:

Which day / session did you like most? (you can describe the device behavior by words if you don't remember the day/session)

What did you like in general? (short text)

What didn't you like in general? (short text)

What would you do differently? (short text)

Bonus - would you buy a device with such capabilities? How much would you pay for it?