DELIVERABLE 2.4

Implementation and integration of context-aware planner for empathic behaviour generation

Author(s):	K.L.Koay et. al.
Project no:	287624
Project acronym:	ACCOMPANY
Project title:	Acceptable robotiCs COMPanions for AgeiNg Years

Doc. Status: Final Draft
Doc. Nature: Deliverable
Version: 1.1
Actual date of delivery: 31/10/ 2014
Contractual date of delivery: Month 36
Project start date: 01/10/2011
Project duration: 36 months
Peer Approver: IPA (FRAUN)

Project Acronym: ACCOMPANY
Project Title: **Acceptable robotiCs COMPanions for AgeiNg Years**

EUROPEAN COMMISSION, FP7-ICT-2011-07, 7th FRAMEWORK PROGRAMME
ICT Call 7 - Objective 5.4 for Ageing & Wellbeing

Grant Agreement Number: 287624

DOCUMENT HISTORY

Version	Date	Status	Changes	Author(s)
1.1	2014-10-22	Final		UH
1.0	2014-10-15	Final draft		UH
0.6	2012-08-20	Draft	Initial Draft	UH

Authors & Contributers

Partner Acronym	Partner Full Name	Person
UH	University of Hertfordshire	K. L. Koay
Contributors:		
UH	University of Hertfordshire	K. L. Koay, D. S. Syrdal, J. Saez, I. Duque, K. Dautenhahn J. Saunders (Reviewer)
IPA	IPA	R. Bormann (Contributor and Reviewer)

Short description

This deliverable reports the on the work outcomes for T2.4 comprising a brief literature review on context-aware systems and human-robot proxemics and the requirements for developing a context-aware planner suitable for empathic behaviour generation for improving the Care-O-bot® proxemics behaviour.

The report details the development of an activity detection system that provides contextual information on the user to the context-aware planner.

In order to support the development of the context-aware planner two user studies exploring robot etiquette in domestic environments as well as robot contingent behaviour were conducted to further our understanding of user's expectations and perceptions of the Care-O-bot® in various interaction situations. Additionally included is the description of the two empathic behaviours proposed in D2.3 for the Care-O-bot®.

The final part of the report summarises the work done in T2.4 and possible future directions for this research.

Table of Contents

1. Introduction

Robots that can adapt their functionality to match their environment and task situation are highly desirable, especially in the field of robotic home companions. Robotic companions not only need to be able to support users with their activities of daily living (ADL) in their home environments, but also need to be socially adaptive by taking into account their users' individual differences, environments and social situations in order to behave in a socially acceptable manner and to gain acceptance into the household. To behave in such a manner while providing ADL support, robots will need to be context-aware, taking account of any contextual information relevant to its services and improve on delivering these services by adapting to the users' requirements.

According to Dey and Abowd (1998), *a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.*

Much research in the field of context-aware systems originates from the field of ubiquitous computing. For example, Marc Weiser (Weiser, 1991) envisioned a scenario in which computational power (of machines) is available anywhere, embedded within the human environment (i.e. walls, chairs, clothing etc.) making information available at our fingertips. This allows for mobile applications to discover and take advantage of contextual information (i.e. time, location etc.) in order to adapt their services to increase usability and effectiveness, without requiring direct user intervention (Baldauf, Dustdar and Rosenberg, 2007).

An example of an early context-aware system was the Active Badge Location System introduced by Want et al. (1992). This system provided user location context to a receptionist operating a switchboard, who could then forward telephone calls to telephones located close to the intended recipient. The emphasis on location information as one of the most useful and widely used attributes of context has continued and led to the development of many location-aware systems such as intelligent tour guides (Abowd et al., 1997, Sumi et al., 1998; Cheverst et al., 2000).

Contextual Information related to location is particularly useful for autonomous systems such as Robotic Companions. User location context is key to many services that a robotic companion can perform, as many of these depend on the robot knowing where the user is as well as how to physically approach the user for interaction, i.e. to offer a drink or provide urgent information.

Context-aware systems are not limited to location-aware systems; Dey et al. (1999) introduced a Conference Assistant system which combines contextual information from both time and location of the users to provide attendees with information related to the presentation that is happening in these locations. Context-aware systems are also widely used in the fields of human-computer interaction, artificial intelligence, computer vision (Crowley et al., 2002) and e-commerce (Palmisano et al., 2008). With no common definition of what is meant by, or included in the term "context", in the field of context-aware systems, different researchers define context differently depending on the specific requirements of a particular context-aware system. Some of these definitions are presented below:

Dey, Salber and Abowd (2001) define context as:

> "*…any information that can be used to characterize the situation of an entity. An entity is a person, a place, or a physical object or computational object that is considered relevant*

> *to the interaction between a user and an application, including the user and applications themselves".*

While this definition has been described as quite vague, Winograd (2001) argues that it is intended to be general enough to cover a variety of research on context-aware interaction. He goes on to argue that:

> *"...something is context because of the way it is used in interpretation, not due to its inherent properties. The voltage on the power lines is a context if there is some action by the user and/or computer whose interpretation is dependent on it, but otherwise is just part of the environment."*

Chaari et al (2006) agree with Winogard's definition and consider context as an operation whose definition depends on the interpretation of the particular operations involved on an entity at a particular time and space, rather than inherent characteristics of the entity itself.

Context instances are categorised into two main context dimensions in the literature. These are external vs. internal (Prekop and Brunett, 2003; Gustavsen, 2002) and physical vs. logical context (Hofer et al. 2002). The external/physical context dimension refers to contexts that can be measured directly by hardware sensors such as location, light, sound, movement, touch, etc. The internal/logical dimension refers to contexts that are specified by the designer and captured or obtained from monitoring user interactions (i.e. the user's goal, tasks, emotional state etc., cf. Baldauf, Dustdar and Rosenberg, 2007).

As context has proven to be useful in other fields, especially for mobile applications, we believe this technology will also be very useful for robotic companions which are intended to interact directly with their users. It is well known that people rely on context to establish the baselines of their interactions in particular with regard to proxemics (Burgoon and Walther, 1990), and as such it would be advantageous for a robot companion to be able to use contextual information for planning and performing its tasks, and thus gain the users' trust for being perceived as socially aware, friendly, intelligent, capable and reliable. This also might help overcome trust issues that may arise with users that may not be familiar with the robot or new technology, such as some elderly people, thus gaining acceptance when inserted into their homes.

With incorporating robot control that uses context, we might be able to minimise some of the safety concerns users might have. This includes concerns about situations when robots may block their path, move behind them or move on a collision path towards them as human and robot interact in a shared space (see discussion of these issues in Koay. et al., 2006, 2007). For these reasons, we focused on developing a context-aware planner for human-robot proxemics.

Contextual information will also provide a robot with the ability to sense their users' interactions with their surroundings and be aware of their activities (i.e. low level activities such as knowing that the user is sitting on a sofa or opening a drawer etc., as well as high level activities such as watching TV or making hot drink etc. (cf. Schmidt, Beigl and Gellersen, 1998; Duque et al., 2012) in order for the robot to take the initiative to proactively support them in their everyday tasks.

The contents of this deliverable are organised in the following chapters:

Chapter 2: Discusses the requirements for the context-aware planner based on the task requirements for the set-up of the environment.

Chapter 3: Discusses the Activity Recognition System we create to contextualise information.

Chapter 4: Presents the two user studies to explore robot etiquette and robot contingent behaviour.

Chapter 5: Presents the development and formative evaluation of the context-aware proxemics planner.

Chapter 6: Discuss the implementation of two empathic behaviours: I See You Seeing Me and Walk with Me.

Chapter 7: Summary and Future Work.

2. Requirements for context-aware planner

In order to support and maintain user independence in their daily lives, the first step when designing a context-aware planner for robotic home companions is to try to understand the users, their everyday lives, and their requirements with respect to the services a robot companion may be able to provide (cf. ACCOMPANY D1.1 and D1.2). Based on this information, we can explore the capabilities of the robot, especially in terms of activities and tasks that the ACCOMPANY robot is capable of carrying out for users in their living environments. The services that can be provided by the robot are not fixed but will be expanded as the understanding of the users' needs progresses, or as new technology becomes feasible for implementation. The current services derived from the scenarios presented in D1.3 can be divided into two main categories: 1) cognitive prosthesis (e.g. reminder) and 2) physical assistance (e.g. fetch and carry etc.).

Generally, cognitive prosthesis involves tasks such as reminding the user of their schedules (i.e. medication, sending a birthday card, telephoning their family etc.) or notifying the user of events within their immediate surrounding that require their attention (i.e. fridge door has been open for 5 minutes, ringing of doorbell etc.).

Physical assistance involves activities such as moving around the user's environment and helping the user carrying objects as well as fetching.

The target for the ACCOMPANY robot is that it should be able to provide a variety of **notifications**, **reminders**, and **fetching** or **carrying** tasks which will support independent living scenarios of users in household environments such as those studied in the UH Robot House, as well as the two other test environments based in the Netherlands and France.

To achieve this target, the ACCOMPANY robot needs to be aware of the activities of the users, their environment and their situation. This contextual information can often be derived from sensors such as those used in smart homes (Kasteren, Englebienne and Kröse, 2010; Chen, Nugent and Wang, 2012; Korpipaa and Mantyjarvi, 2003). Raw sensory data from these sensors can be converted into meaningful semantic symbolic expressions that can then be used to describe activities of the users, events in their environment, or their overall situation. These semantic symbolic expressions can be as simple as an action performed by the user, such as sitting down, which can be directly detected from the appropriate sensors without further processing. On the other hand, they can be as for example, a making-a-cup-of-tea activity, which is not directly detectable from sensors, but could be derived by combining different user actions within a particular time frame (i.e. accounting for the process of making tea). Together these semantic symbolic expressions form the main mechanism that provides the ACCOMPANY robot with the contextual information needed for it to perform its tasks. This contextual information can be divided into the following five different categories (i.e. one physical context and four logical contexts) taking inspiration from Mostefaoui and Hirsbrunner (2003):

> **Physical Context:** Contexts that can be measured directly from hardware sensors i.e. drawer is open/closed, doorbell is ringing, light, weather, temperature etc.
>
> **User Context**: User activity, user location, user role, user preferences, user social situation and user permission profile etc.

Robot Context: Robot activity, robot location, robot role.

Time Context: Current time, day, year, month and season etc.

Context History: A time–stamp log of the above contexts which can be used to improve the robot system.

Using the contextual information presented above, the robot could in principle know when to take the initiative in assisting its users as well as taking into account the users' preferences and overall social situations within these interactions. For example, the robot would know when to remind users about their medication, or notify users if someone is at the door or make them aware that the fridge door has been left open for too long.

To make this contextual information available to the robot, the UH Robot House is equipped with two commercially available sensor systems, the Green Energy Options (GEO) Trio System and the ZigBee (ZigBee) Sensor Network. This setup provides over 50 sensors, targeting activities at relevant location, such as the Dining Area, Living Room, Kitchen, Bedroom and Bathroom of the UH robot house (see Figure 1).

The GEO System is a real-time electrical device energy monitoring system and is used in the UH robot house to detect the activation and de-activation of specific electrical appliances by the user, such as when the refrigerator is opened, water is boiled in the kettle, or detecting when the doorbell has been pressed in the case of visitors at the door. The Zigbee Sensor Network is a standards-based (Xbee GatewayX4) low-power wireless sensor system. It is used in the UH robot house to detect user activities that cannot be detected by the GEO System, such as the opening and closing of drawers and doors, occupancy of chairs and sofa seat-places, water being run through taps in the kitchen and bathroom etc. The three main sensors types currently installed are Reed Contact Sensors, Pressure Mat Sensors and Temperature Sensors. An Activity Recognition System has been created to interpret these data to convert them into meaningful contextual information. This is described in the next section. Note, the sensor network is a system that has been used and tested extensively in a previous FP7 project (LIREC, 2008-2012). It was used for example in two long-term studies whereby in total 208 one-hour sessions were carried out involving 20 adult participants as part of long-term studies. In ACCOMPANY this system has been extended to include logical context for recognize complex human activities that involves more than one physical contextual information such as making a cup of tea, preparing a meal etc. During the duration of project other sensors and activity recognition processes have been added to build a repertoire of user context for used in the Memory Visualisation and the TeachMe/ShowMe Systems (see deliverable D3.4 for more details of these systems).

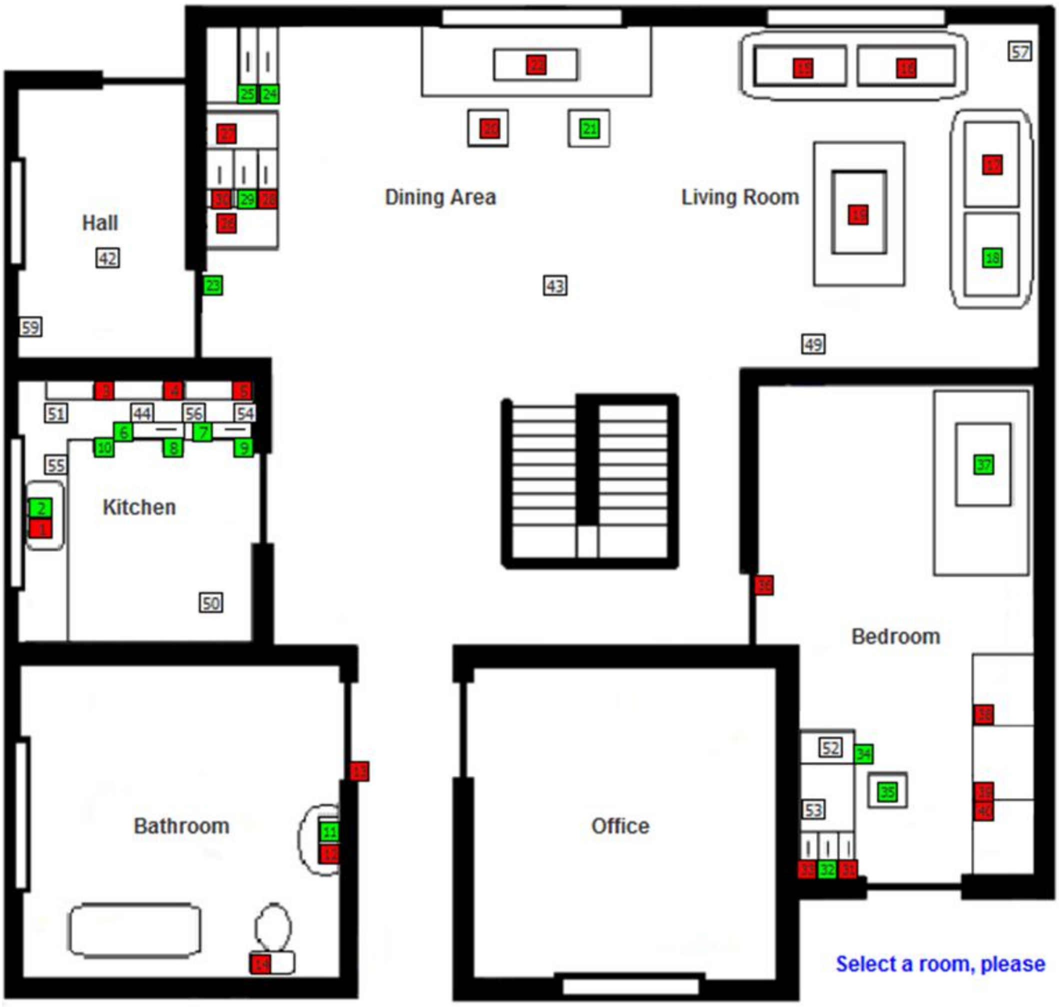


Figure 1 - UH Robot House map showing the location of sensors (identified by numbers) and their states with green colour representing the sensor in open/on/free state, red colour representing the sensor in close/off/occupied state and transparent representing the sensor is not activated/unknown.

3. Rule-based Activity Recognition System

This chapter presents the works conducted in year 1 on a knowledge-driven, rule-based Activity Recognition System (ARS) (Duque et al., 2012) developed to derive sensory information from both the GEO System and the Zigbee Sensor Network embedded in the UH robot house. This formed the baseline requirement for the development of the context-aware planner. This system has now been partially superseded by work done on the TeachMe/ShowMe system reported in D3.4 (which used data derived from the ARS system to evaluate the feasibility of the TeachMe/ShowMe system) as well as the vision based activity recognition system reported in D4.4 The ARS system originally was designed to provide the contextual information necessary to guide the robot's social behaviour.

The reasons for selecting a knowledge-driven approach over probabilistic/machine learning approaches (as they are used in WP4) for the Activity Recognition System were to: a) avoid necessity to collect large amounts of training data from elderly users, b) provide a flexible approach so that the detectable user activities can be easily extended and modified during the development and fine tuning of the context-aware planner and the ACCOMPANY scenario, c) create a system that is easy to install and setup in other similar environments without the necessity of specialised knowledge (since the rules are based on a natural language description and are explicitly represented, rather than the implicitly representation e.g. within a Bayesian network (Tapia, Intille and Larson, 2004; Bao and Intille, 2004) or a Hidden Markov Model implementation (Sanchez, Tentori and Favela, 2008; van Kasteren ea al., 2008)). The Activity Recognition System presented here will complement the work done in WP4 where machine learning approaches have been investigated, cf. (Kasteren, Englebienne and Kröse, 2010; Hu, Englebienne and Kröse, 2014, which work on other kinds of data, such as body movements, object locations, or relational features, derived from vision sensors. As a consequence, that system can recognize different kinds of activities at different levels of detail than ours.

Currently, the Activity Recognition System is able to detect user activities directly from single sensor data (without the need to fuse data from different sensors) and predict user activities attached through a knowledge-driven approach that combines contextual information (based on sequences of activities, or activities performed concurrently by the user or the robot) and sensor data. Rules for detecting each user activity can be set up by filling in the required conditions in the appropriate field in the rule file. A skeleton rule file is shown in Figure 2.

```
<Activity Name=“”>
  <Duration></Duration>
  <Location=“”>
  <Contexts>
    <Context Interval=“” Status=“”> </Context>
  </Contexts>
  <Sensors Threshold=“”>
    <Sensor Status=“” NotLatching=“” Weight=“”> </Sensor>
  </Sensors>
</Activity>
```

Figure 2 - A skeleton xml descriptor for defining the user activity detection rules.

The rules for detecting activities are defined using the following tags:

Activity Name – the name of the *New Activity* this rule file is for.

Duration – the duration that the *New Activity* remains activated for, after it is detected. This only applies to activities for which the system cannot detect deactivation. Activities such as *Using_Computer_Dining_Area* or *Sitting_Living_Room* do not require this tag as the system is able to detect the deactivation of these activities via their associated sensors or contextual activities.

Location – the name of the location where the *New Activity* will take place.

Contexts – contains a list of contextual activities that have to be fulfilled before the *New Activity* can be considered as detected or considered as one of the candidates for the detected activity. Activities such as *Sitting_Living_Room* do not required any contextual activities associated with them as they can be directly detected from the sensory networks.

Context – the contextual activity relevant for the detection of the New Activity.

Interval – is the time window which a context activity state remains valid/relevant for the detection of the new activity.

Status – defined the required context activity’s state.

Sensors – contains a list of the sensors conditions to be satisfied before the New Activity can be considered as detected.

Threshold – minimum accumulated sensor weights needed for the activation of this new activity.

Sensor – the sensor relevant for the detection of the New Activity.

Status – define the required sensor’s state.

Weight – define how important this particular sensor state is for the detection of the *New Activity*.

NotLatching – (true) the sensor weight will only be added to the accumulated weight while it remains on, (false) the sensor weight is added once it’s on regardless of its state after.

Figure 3 shows an example of the rules that define user activity based solely on sensory data. The *Sitting_Living_Room* activity is associated with the sensors attached to the sofa in the living room. In this example, the Sensors’ *Threshold* tag is set to 0.2, and each sensor has a weight of 0.2 when they get turned on (i.e. when the user sits on it). If the sensor turns on, this user activity *Sitting_Living_Room* is detected. Note that *NotLatching* tag is set to true since we want to deactivate this activity as soon as the user is no longer sitting on the sofa. The *Duration* tag is set to nil as it is not needed for this activity, since the deactivation of the associated sensors can be detected directly to deactivate the activity.

```
<Activity Name=“Sitting_Living_Room”>
  <Duration>Nil</Duration>
    <Location>Living_Room</Location>
    <Contexts></Contexts>
    <Sensors Threshold=“0.2”>
      <Sensor Status=“on” NotLatching=“true” Weight=“0.2”> Sofa seatplace 0</Sensor>
      <Sensor Status=“on” NotLatching=“true” Weight=“0.2”> Sofa seatplace 1</Sensor>
      <Sensor Status=“on” NotLatching=“true” Weight=“0.2”> Sofa seatplace 2</Sensor>
      <Sensor Status=“on” NotLatching=“true” Weight=“0.2”> Sofa seatplace 3</Sensor>
      <Sensor Status=“on” NotLatching=“true” Weight=“0.2”> Sofa seatplace 4</Sensor>
    </Sensors>
</Activity>
```

Figure 3 - Example of Sitting_Living_Room rule file.

Figure 4 shows an example rule set that defines a user’s activity based on contextual information. The *Using_Computer_Dining_Area* activity depends only on *Sitting_Dining_Area* and *Computer_On* activities to be activate. Therefore the Sensors tag does not contain any conditions and the *Threshold* tag is set to 0.0. The *Duration* tag is not needed for this activity and is set to Nil.

```
<Activity Name=“Using_Computer_Dining_Area”>
  <Duration>Nil</Duration>
  <Location=“Dining_Area”>
  <Contexts>
    <Context Interval=“0” Status=“activated”> Sitting_Dining_Area </Context>
    <Context Interval=“0” Status=“activated”> Computer_ON </Context>
  </Contexts>
  <Sensors Threshold=“0.0”></Sensors>
</Activity>
```

Figure 4 - Example of Using_Computer_Dining_Area activity rule file.

Through the rules file, the Activity Recognition System can easily be updated for different environments, and new user activities can be added as the sensor networks are improved. Table 1 lists some of the activities the system is currently able to detect or predict. Details of the Activity Recognition System design, implementation and evaluation can be found in Duque et. al. (2012, submitted).

Table 1: Example of user activities that can be detected by the Activity Recognition System

Location	**Low Level Activity -** user activities directly detectable from sensory information.	**High Level Activity -** user activities that can be derived from fusion of current sensory information and contextual information from both the user's previous activities or initiated by the robot.
Dining Area	-turning computer on/off -sitting on the chair	-using computer -reading a book/Newspaper -writing letter/birthday card etc. -having meal -cleaning table -playing game
Kitchen	-using microwave -using toaster -using kettle -using dishwasher -using kitchen's taps -opening the fridge -opening the cattery drawer	-preparing food -preparing cold drink -boiling water/making hot drink -cleaning dishes -drinking water or cleaning
Living Room	-turning TV on/off -sitting on the sofa	-watching TV -playing game
Hall	-doorbell ringing	-newspaper delivery

The implementation of the Activity Recognition System in the robot house currently allows the ACCOMPANY robot to take the initiative and therefore is hoped to provide a better interaction experience for the users. Future user studies need to confirm the acceptance of this new feature.

A mechanism that allows users to teach the robot to generate and take advantage of activities defined by themselves at higher semantic levels was developed as part of the work done in WP3 and described in deliverable D3.4.

4. User Studies

Two user studies were conducted as part of WP2.4 in the second year of the ACCOMPANY project to explore robot etiquette in domestic environment Koay et al. (2013) and to further the understanding how people perceived contingent behaviours exhibited by the Care-O-bot® Saez-Pons et al. (2014).

We will briefly describe the results of these studies here. Further discussion and detailed information regarding these studies can be found in their respective publication (Koay et al., 2013 and Saez-Pons et al., 2014).

Exploring Robot Etiquette

The main aim of this study was to explore robot etiquette, focusing on understanding behaviours that people might expect from a robot that lives and shares space with them in their home. The experiment was intended to tease out passive behaviours that can be added to the robot's active behaviours to make the robot appear more considerate and socially intelligent. In addition, the results were also to highlight context-aware human-robot proxemics for Care-O-bot and its spatial configurations for a given interaction. This information was later used for improving the context-aware planner discussed in Chapter 5.

The experiment was conducted with two residential artists at the UH Robot House. At the time of the experiment both artists had spent time in and habituated to the robot house and were familiar with its domestic setting. This made them able to provide insightful feedback related to human-robot space negotiation in a domestic environment.

The artists would interact with the robot one at a time, and shown the scenario inspired by the Accompany Project Year 1 Scenario, which involved the Care-O-bot® reminding and assisting an elderly user to fetch a bottle of water, in a step-by-step walk-through. At different stages within the scenario that involved the robot, the participant (the artist) was asked specific questions about the robot's behaviour and their feedback was recorded.

The questions used were in the form of:

i. *How should the robot perform the particular task?*
   a) Should the robot make any sounds? If so, what kind of sounds?
   b) Should the robot display something on its LED colour display? If so, what colour or display behaviour should it exhibit?

ii. *Where and how should the robot position itself for this part*?
   a) Position, orientation and posture.
   b) Reasoning behind this decision.

The robot was present throughout the study and was remotely controlled by the experimenter in order to demonstrate its functionality and to act as a reference for the participants' creation of robot behaviours at the different stages. Figure 5 shows a picture taken during the experiment where the experimenter and the artist were discussing the scenario.

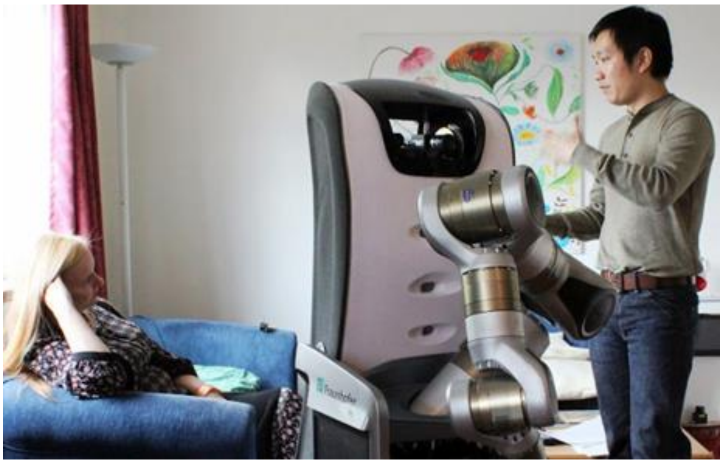


Figure 5 - The experimenter guiding the artist through the scenario.

Results from this study, shown in figure 6, demonstrate the new robot behaviours, which incorporated suggestions and concerns from the artists' feedback, within the experimental scenario. For details about each individual artist's preferences please refer to Koay et al., 2013.

Starting from Fig 3-a) where the robot is in charging mode (its LED colour display shows an amber colour). Here, the robot makes a short beep, changes its LED colour display from amber to flashing green and starts moving towards the user (See Fig 3-b). Note that it takes human-robot proxemics [Walters et al., 2005, 2006, 2007, 2009; Koay et al., 2009; Takayama and Pantofaru, 2009] into account when approaching the user.

Its LED colour display changes from flashing green (navigating) into a solid green as the robot stops moving, and starts flashing blue as it tilts its head forward and reminds the user to have a drink using speech (see Fig 3-c).

If the user agrees to go to the kitchen with the robot to fetch a bottle of water, the robot's LED colour display starts flashing green as it slowly makes room for the user. The robot takes an "after you" posture and changes its LED colour display to solid green (see Fig 3-d and 3-e).

Then the user gets up and starts walking towards the kitchen while the robot displays flashing green LED colour and follows the user from behind (see Fig. 3-f).

As the user opens the fridge to fetch a bottle of water, the robot slowly approaches the user taking up a position close to the user, but not blocking the kitchen entrance. It lifts its tray, as a gesture to offer assistance to carry the bottle, and then switches its LED colour display from flashing to solid green (see Fig 3-g).

The user then places the bottle on the robot's tray and the robot's LED colour display starts flashing green. The robot slowly moves back to make way for the user. It takes an "after you" posture and stops flashing its LED colour display (see Fig 3-h).

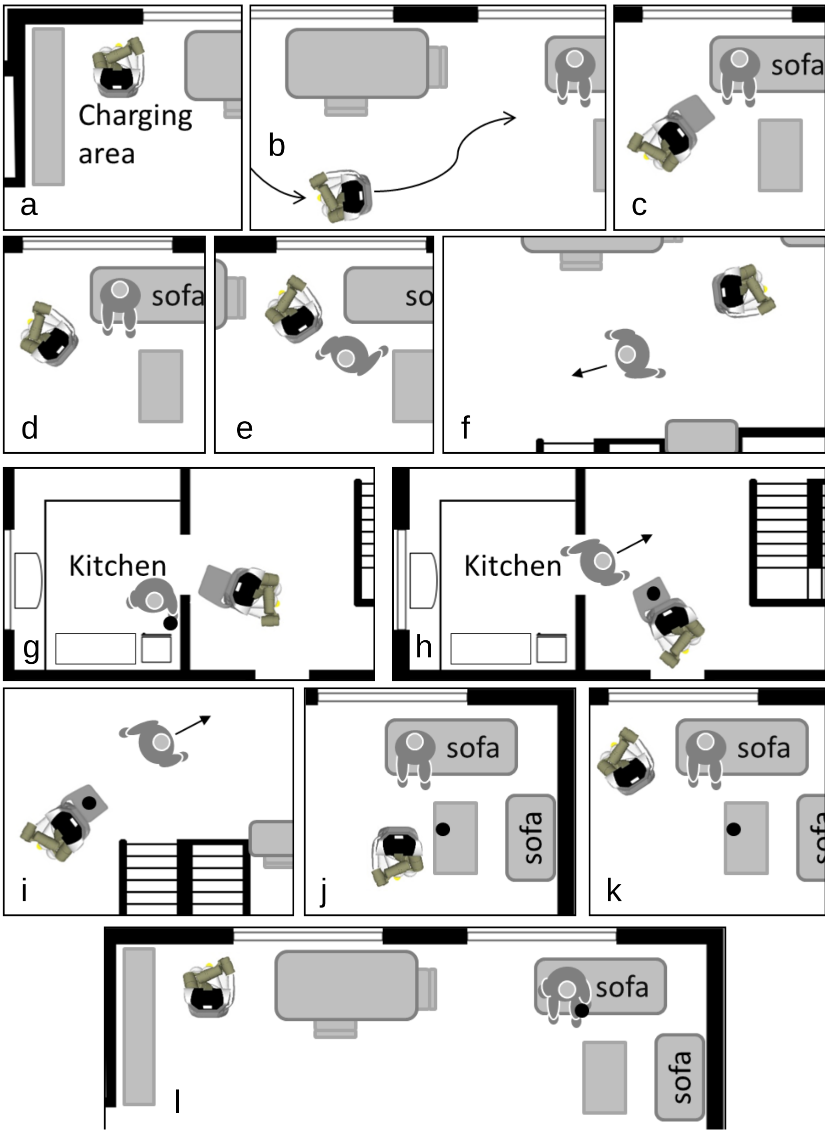


Figure 6 - How the new robot behaviours, created by integrating the artists' feedback, will act in the scenario.

The user starts walking towards the sofa area, while the robot LED colour display starts flashing green and the robot follows the user from behind (see Fig 3-i).

The user sits on the sofa while the robot slowly approaches and stops next to the table. It then keeps its distance from the user while switching its LED colour display from flashing green to flashing red. The robot moves its arm slowly to grab the bottle from its tray and place it on the table. It then parks its arm at its back, as shown in Fig 3-j, then switches its LED colour display from flashing red to flashing green.

The robot then moves to take up a position next to the user, taking into consideration of not blocking the user's view of the television is located in the living room, or from other areas of the experimental area (see Fig 3-k).

The robot switches its LED colour display to solid green and stays there, waiting to remind the user to have their drink if they forget. After the user drinks the water and the robot has no immediate task to interact with the user for, it will switch its LED colour display to flashing green and then slowly navigate back to the charging station.

The robot will then go into charging mode with its LED colour display showing amber (see Fig 3-l).

An overview of the key behaviours according to tasks is provided in Table 2.

Table 2: An overview of the key robot behaviours.

Robot Task	Robot Behaviour
Charging	Its LED colour display shows amber colour.
Navigation	Makes a short beep, changes its LED colour display to flashing green and starts moving towards the user.
Approaching and reminding a seated user	Takes human-robot proxemics into account when approaching the user. Changes the LED colour display from flashing green (navigating) into a solid green as the robot stops moving.
Speaking	LED colour display flashes blue, tilts its head forward then speaks.
Giving way to the user	LED colour display flashes green as it slowly makes room for the user. The robot takes an “after you” posture then changes its LED colour display to solid green as it stops moving.
Following	The robot LED colour display flashes green as it follows the user from behind.
Approaching a user standing by an entrance	Robot slowly approaches the user to taking up a position close to the user, but not blocking the entrance.
Offering assistance	Lifts its tray, as a gesture to offer assistance to carry the bottle. Then switches its LED colour display from flashing to solid colour.
Placing an object on the table next to the user	Robot slowly approaches and stops next to the table. Then keeps its distance from the user while switching its LED colour display to flashing red. The robot then moves its arm slowly to grab the bottle from its tray and places it on the table. It then parks its arm at its back, then switches its LED colour display to flashing green.
Accompanying the user	Robot moves to take up a position next to the user, taking into consideration of not blocking the user’s view of the television located in the living room or other important areas of the experimental area.
No immediate task	It switches its LED colour display to flashing green, then slowly navigates to the charging station.

Contingent Behaviours

The main aim of the contingent behaviours study was to understand how synchronised movements of a non-anthropomorphic robot influence the user’s perception of the robot and what role the direction of synchronisation plays. The study was based on the idea that robots exhibiting synchronized human-like movements such as head gaze following may induce an emotional reaction in the user, even when the robot does not have a clearly distinguishable head such as the Care-O-bot®. We believe that the robot would be perceived as more friendly and likeable if it exhibits behaviour movements in positive synchronisation with the actions of the users.

The study (Saez-Pons et al, 2014) was conducted using an online survey methodology where participants from various mailing lists (i.e. the robotics-worldwide, euron-dist and PHILOS-L) were invited to participate. Participants were invited to watch three different videos of an active human user arranging flowers lying on the table into a bouquet while the Care-O-bot® robot is watching from the opposite side of the table. The three different videos were produced to represent the three different experimental conditions explored in this survey.

The first condition shown in Figure 7a was the control condition where the robot was not moving at all. Its torso was fixed facing the user throughout the experiment for this condition.

The second condition involved the Care-O-bot® moving its upper torso exhibiting negative synchrony. Negative synchrony illustrated that the robot torso movement was synchronised with the user's movement, however the robot's torso movement was always to the opposite direction of the user's movement. This gave the impression that the robot was avoiding or not engaged with the user's task (see Figure 7b and Figure 8a).

The third condition involved the Care-O-bot® moving its upper torso in positive synchronisation with the user's movements (i.e. direction of action). It synchronised its movement with the user's movements and followed the user's actions towards the objects on the table accordingly, giving an impression of joint attention and engaging in what the user was doing (see Figure 7c and Figure 8b).

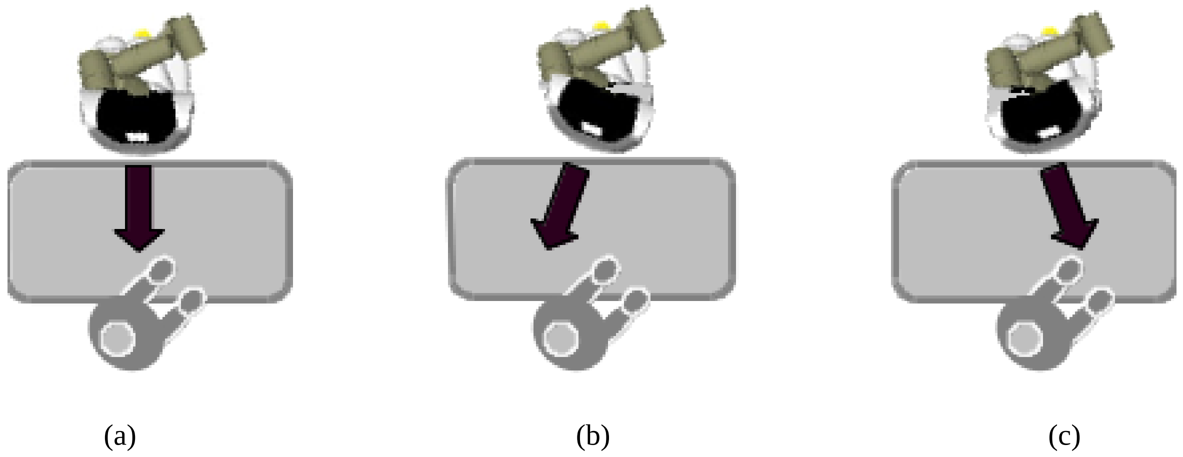


Figure 7 - The three conditions used in the Contingent Behaviour study: a) the control condition with static robot, b) the negative synchrony condition, and c) the positive synchrony condition.

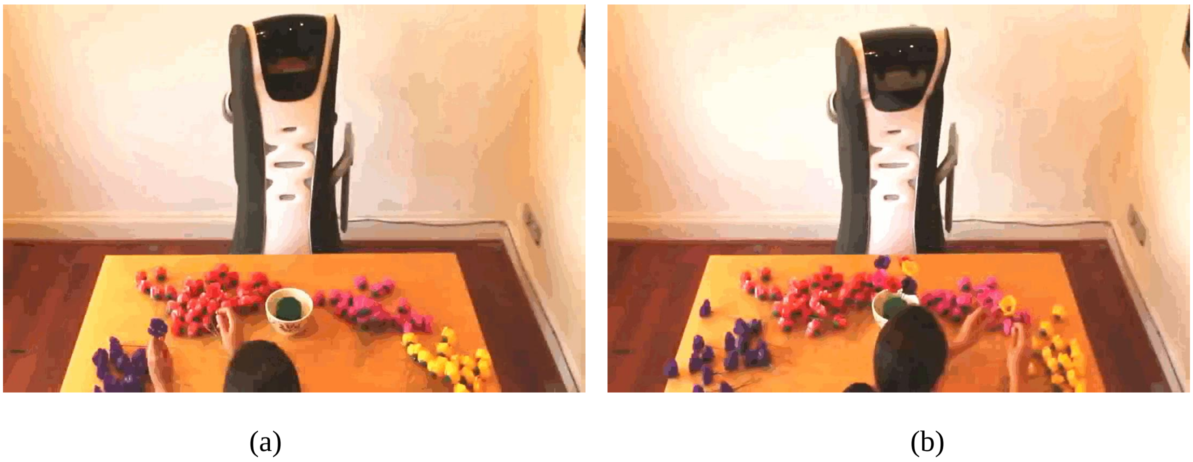


Figure 8 - Video stills from the final validation online video study showing (a) negative and (b) positive synchrony conditions.

The participants watched and filled in a questionnaire for each of the videos which were presented in a randomised order.

The main result from the study indicated that the Care-O-bot® behaviour shown in the videos did induce a reaction from the participants, and their rating for all three conditions differed from one another.

The results suggested that the Care-O-bot® that exhibited a positive synchronisation with the user's actions was rated the most likeable and intelligent, followed by the condition in which the robot exhibited negative synchronisation. The Care-O-bot® from the third condition, which did not move at all, was rated the least likeable and intelligent.

This indicates that synchronised movements is a powerful tool that can be used to communicate intention in a similar way to that of a living being using their head or eyes movements to communicate their intentions. The study also demonstrates that synchronised movements can be used to make robots that do not have an anthropomorphic head or eyes employ synchronised movements through other body parts (i.e. torso or whole body movement) to communicate intention. Therefore, the positive synchrony condition makes Care-O-bot® appear more intelligent and likeable when compared to other experimental conditions.

5. Implementation of Context-aware Planner

Context-aware Planner

The context-aware planner presented here aims to improve the Care-O-bot®'s social behaviour by adapting its distances and orientation in terms of interpersonal space, based on the contextual information of the user and the robot.

Research (Walters et. al., 2005, 2006; Koay et al., 2007; Takayama and Pantofaru, 2009) has shown that proxemics (how interactants negotiate interpersonal space within an interaction) play an important role in human-human interactions as well as those between human and robots. Therefore it is essential that the Accompany System is able to take into account the users' proxemics preferences when approaching them for interactions.

According to the literature (Walters et. al., 2005, 2006, 2009; Koay et al., 2007), users' proxemics preferences vary depending on their familiarisation/experience with robots, situation and the context of the interaction. For example, a robot approaching a person who is seated in the living room with the aim of interacting with the user should behave differently depending on the activity the person is engaged in and the purpose of the robot initiated interaction. If the user is watching TV in the living room, they may not want the robot to approach and stop at their preferred (relative) approach position and orientation as it might block their view of the TV. However, this approach, and interruption, may be appropriate if the robot is presenting urgent information that needs to be acted upon, such as a visitor at the door.

The context aware planner proposed in the project aimed to improve the robot's proxemics behaviour by providing appropriate target coordinates for the robot to approach the user in a socially acceptable manner, and to maintain a suitable interaction distance from the user. It ensures that the robot will always have a solution to approach the user as close as possible even when the user is in a small confined area or in an area in which the robot may not have access to get close to the user.

The planner was built as a ROS service which allows a client to call the service by sending the request message and awaiting the reply. The planner was designed to be a separate module that is independent from the Care-O-bot's navigation system (i.e. costmap, path planner etc.). Therefore, it can be used with different navigation systems on different robotic platforms with minimum reconfiguration or modification. For the purpose of debugging, testing and conducting technical evaluation, a standalone client was provided to allow an experimenter to call the context-aware planner service from the terminal. To do this, the client needs to provide the following parameters: the user's id, posture, coordinate (x, y, theta) in the map, and the robot's task at the target coordinate.

Replies from the context-aware planner service are in the form of ranked target coordinates. The ranked target coordinates are sent to the Care-O-bot's navigation system one by one, depending on whether the robot can reach the previous target coordinate (i.e. if the robot fails to approach the first target coordinate due to unexpected obstacles, the second target coordinate will be sent). This process will continue until the robot reaches one of the target coordinates.

In the ACCOMPANY project, the context-aware planner's client is part of the UHCore python library developed during the project to provide an interface for the ACCOMPANY system to have direct access to the Care-O-bot®'s sensors and actuation modules. For example, when the COBCoreScheduler (the robot behaviour control system developed in WP3) issues a request to navigate to the user's location, it is the responsibility of UHCore to send this request message to the context-aware planner in order to obtain suitable target coordinates and ensure that the Care-O-bot® successfully approaches the user by sending the target coordinates one at a time to the Care-O-bot® navigation system until the navigation system reports back that the robot has successfully reached the given target coordinate.

The context-aware proxemics planner consists of three components, here we will provide an overview of each components and discuss how these components overcome some of the issues encountered in a domestic environment to ensure that the robot will always have an appropriate set of target coordinates for approaching the user in a friendly manner for interaction. These three components are: i) General Proxemics Preferences Based Algorithm, ii) Exceptional Cases Preferences Algorithm, and iii) Location Ontology Based Algorithm.

General Proxemics Preferences Based Algorithm

The General Proxemics Preferences Algorithm was inspired by the literature from human-human proxemics and from human-robot proxemics studies [Walters et. al., 2006; Koay et al., 2007] in the field of Human-Robot Interaction.

It uses the user's coordinates to generate a maximum set of 21 possible target coordinates around the user. Visually these coordinates are arranged in 3 layers of three quarter-circle perimeter configurations where the distances of the layers from the user are at 0.4m, 0.7m and 1.5m respectively (see Figure 9). These distances were adapted from the literature discussed above. 0.4m and 0.7m are reserved for physical interaction such as performing fetch and carry tasks for the user, while 1.5m is reserved for verbal interaction such as providing notifications or reminders to the user. The default distance for physical interaction is 0.7m while 0.4m is reserved for users that have experience with interacting with similar robots.

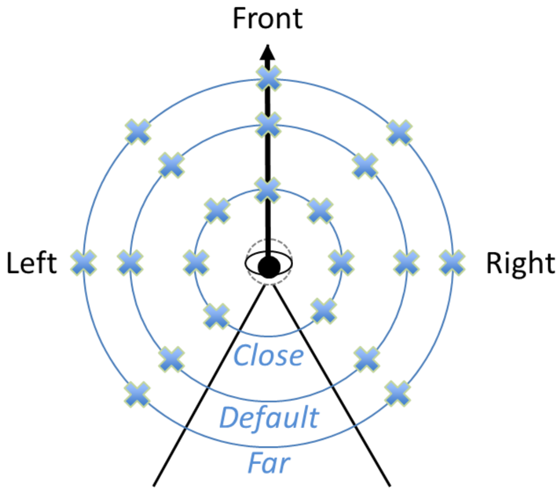
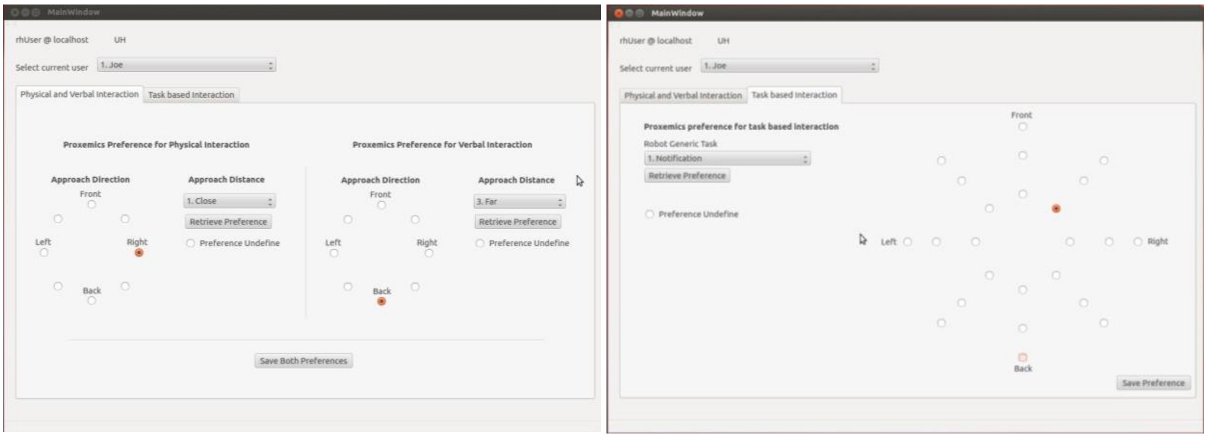


Figure 9 - Example of possible target locations around the user.

Each layer consists of 7 target coordinates which are arranged 45 degrees apart from each other in right back, right side, front right, front, front left, left, and left back directions relative to the user's perspective. Depending on the user's proxemics preferences, or their experiences with robots and the robot's task, the generated 21 possible target coordinates will be ranked accordingly in the dimension of user friendliness, so that the Care-O-bot® can approach the user using the highest ranked (more human friendly) target coordinate and only use a lower ranked coordinate when a higher ranked coordinate is not accessible due to obstacles. Note that the target coordinates will be ranked differently depending on the user's proxemics preferences, their experience or familiarity with similar robots and the type of task (i.e. physical or verbal interaction) the robot will perform at the target coordinates.

The user can personalise their preferences using the GUI personalisation tool shown in Figure 10. The GUI personalisation option for the user to personalise the robot's proxemics behaviours based on the types of human-robot interaction they are going to engage with the robot (i.e. physical or verbal interaction) or based on specific task the robot is going to perform for them (i.e. notification, reminder, fetch and carry with tray etc.)



(a)

rhUser @ localhost UH

Select current user: 1. Joe

Approach Distance (Physical Interaction): 1. Close

Approach Distance (Verbal Interaction): 3. Far

○ Preference Undefine

(b)

rhUser @ localhost UH

Select current user: 1. Joe

Robot Generic Task: 1. Notification

○ Preference Undefine

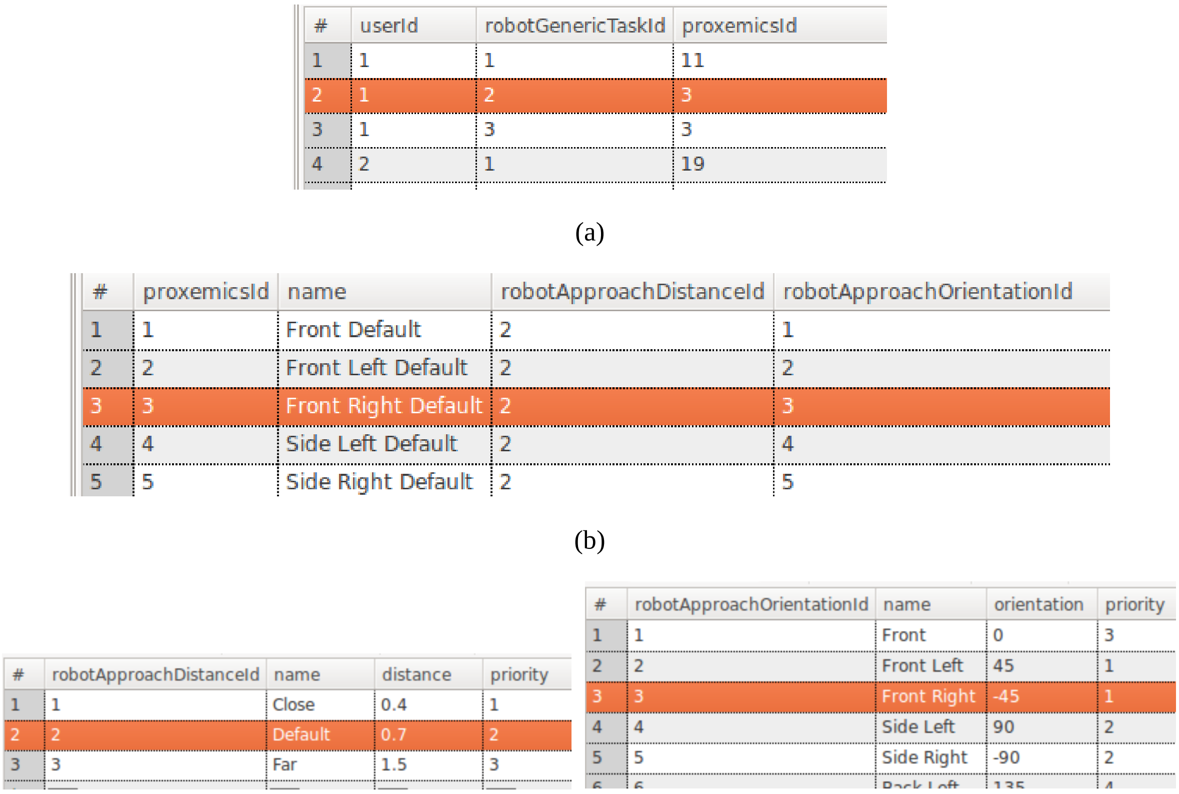
Figure 10 - Proxemics Personalisation GUI for personalisation based on: a) general interaction type, and b) specific robot task.

To request for ranked target coordinates from the context-aware planner, the client request must provide the user's id, posture, coordinates and the robot task at the target coordinates. Using the user's id, and robot task information the planner can then retrieve the user's proxemics preferences (i.e. proxemicsId) from the ACCOMPANY MYSQL database.

The retrieval process begins by identifying the user's preferred *proxemicsId* from the **UserProxemicPreferences** table (see Figure 11a). With this information, the system then retrieves the proxemics configurations for the *proxemicsId* from the **Proxemics** table (Figure 11b). Using the information in the **Proxemics** table, the system can now retrieve the parameters for the configurations from the **RobotApproachDistance** table (Figure 11c) and the **RobotApproachOrientation** table (Figure 11d) such as the preferred distance the robot should maintain and the preferred angle the robot should approach from respectively that is needed to calculate and rank a set of possible target coordinates for the robot.

The ranking algorithm uses the priority data from the **RobotApproachDistance** table to determine the priority of the distance layers, followed by using the priority data from the **RobotApproachOrientation** table to rank the target coordinates in that layer.

For example, if the user prefers the robot to approach from their front right with a stopping distance of 1m for a fetch and carry task, *proxemicsId 3*, the algorithm will place *proxemicsId 3* with the highest priority on its list, followed by its immediate neighbour on the same side (i.e. right side), in this case the right side with priority 2 will be ranked 2nd, followed by front with priority 3 and finally right back with priority 4 before proceeding to the other side (left side) of that layer. Since the priority of front left is one, it will be ranked next followed by the rest of the coordinates on that left hand side. The next distance layer to follow is the Close layer since it has the priority of 1. The coordinate ranking for this layer will be based on the order of the previous layer. This is then follow by the last distance layer which has the lowest layer priority, again the coordinate ranking for this layer will be based on the previous layer. Using this method, the planner will be able to provide 21 ranked target coordinates around the user for the robots.



#	userId	robotGenericTaskId	proxemicsId
1	1	1	11
2	1	2	3
3	1	3	3
4	2	1	19

(a)

#	proxemicsId	name	robotApproachDistanceId	robotApproachOrientationId
1	1	Front Default	2	1
2	2	Front Left Default	2	2
3	3	Front Right Default	2	3
4	4	Side Left Default	2	4
5	5	Side Right Default	2	5

(b)

#	robotApproachDistanceId	name	distance	priority
1	1	Close	0.4	1
2	2	Default	0.7	2
3	3	Far	1.5	3

#	robotApproachOrientationId	name	orientation	priority
1	1	Front	0	3
2	2	Front Left	45	1
3	3	Front Right	-45	1
4	4	Side Left	90	2
5	5	Side Right	-90	2
6	6	Back Left	135	4

(c) (d)

Figure 11 - Tables in the Accompany database that store users' proxemic preferences and the data necessary to compute lists of ranked target coordinates.

These ranked target coordinates then go through an elimination process to verify that they are valid at the user's location before being sent to the robot. The elimination process involves verifying these coordinates with a static map of the environment to ensure that they are (a) in the same location as the user, (b) the location can be occupied by the robot and that (c) the coordinates are reachable by the robot from its current location. Only the target coordinates that survive this elimination process are sent back to the client.

The client can then send the first target coordinate to the robot and only send the rest in the order of one after another if the robot fails to reach the previous target coordinate due to dynamic obstacles etc. in the environment.

Figure 12 shows an example where the valid ranked target coordinates were plotted around the user who is sitting at the location marked X. Note the most preferred target is marked with a darker coloured arrow than a least preferred target which is marked with a lighter coloured arrow.

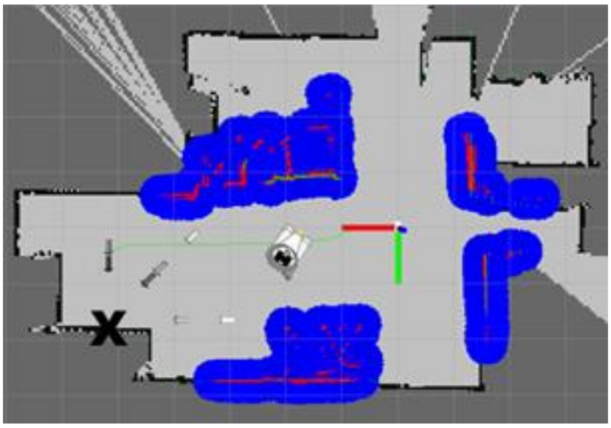


Figure 12 - An example of valid target coordinates (arrow) around the user location marked X.

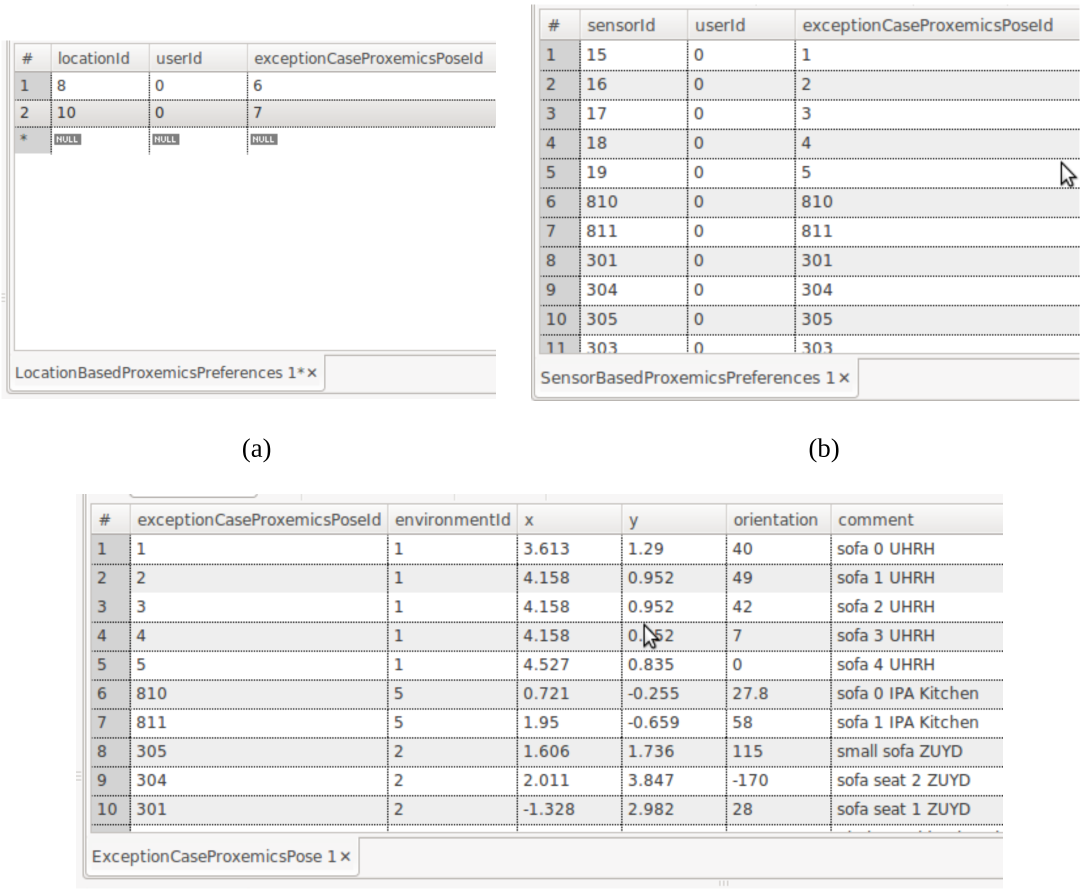
Exceptional Cases Proxemics Preferences Based Algorithm

The Exceptional Cases Proxemics System deals with special cases such as specific preferences for specific situations or locations. For example, the user can specify that the robot should be in a specific location when the user is watching TV or when the user is in the kitchen. There are two ways the user can set these preferences; one is to set the preferred coordinate to be based on the user's location, the other is to set the preferred coordinate based on the activity of one or more specific sensors at the user's location. The Exceptional Cases Preferences algorithm will then utilise the contextual information of the user's location or that of the sensors triggered by the user to send the robot to these specific locations. The user preferences for exceptional cases are stored in the following 3 tables in the Accompany database (see Figure 13):

LocationBasedProxemicsPreferences table: this table stores the exceptionCasesProxemicsId based on locationId for all the users. The algorithm will search this table to determine if there is any exceptionCasesProxemicsId set by the user for their location. If an exceptionCasesProxemicsId is found, it will then retrieve its coordinates from the ExceptionCaseProxemicsPose table.

SensorBasedProxemicsPreferences table: this table stores the exceptionCasesProxemicsId based on sensorId for the user. The algorithm will search this table to determine if there is any exceptionCasesProxemicsId set by the user for any sensors triggered at their location. If an exceptionCasesProxemicsId is found, it will then retrieve its coordinates from the ExceptionCaseProxemicsPose table.

ExceptionCaseProxemicsPose table: this table stores all the coordinates for exceptionCasesProxemicsId for different environmentId (different environment). The algorithm uses this table to retrieve target coordinates for exceptionCasesProxemicsId.



#	locationId	userId	exceptionCaseProxemicsPoseId
1	8	0	6
2	10	0	7
*	NULL	NULL	NULL

LocationBasedProxemicsPreferences 1* ×

(a)

#	sensorId	userId	exceptionCaseProxemicsPoseId
1	15	0	1
2	16	0	2
3	17	0	3
4	18	0	4
5	19	0	5
6	810	0	810
7	811	0	811
8	301	0	301
9	304	0	304
10	305	0	305
11	303	0	303

SensorBasedProxemicsPreferences 1 ×

(b)

#	exceptionCaseProxemicsPoseId	environmentId	x	y	orientation	comment
1	1	1	3.613	1.29	40	sofa 0 UHRH
2	2	1	4.158	0.952	49	sofa 1 UHRH
3	3	1	4.158	0.952	42	sofa 2 UHRH
4	4	1	4.158	0.952	7	sofa 3 UHRH
5	5	1	4.527	0.835	0	sofa 4 UHRH
6	810	5	0.721	-0.255	27.8	sofa 0 IPA Kitchen
7	811	5	1.95	-0.659	58	sofa 1 IPA Kitchen
8	305	2	1.606	1.736	115	small sofa ZUYD
9	304	2	2.011	3.847	-170	sofa seat 2 ZUYD
10	301	2	-1.328	2.982	28	sofa seat 1 ZUYD

ExceptionCaseProxemicsPose 1 ×

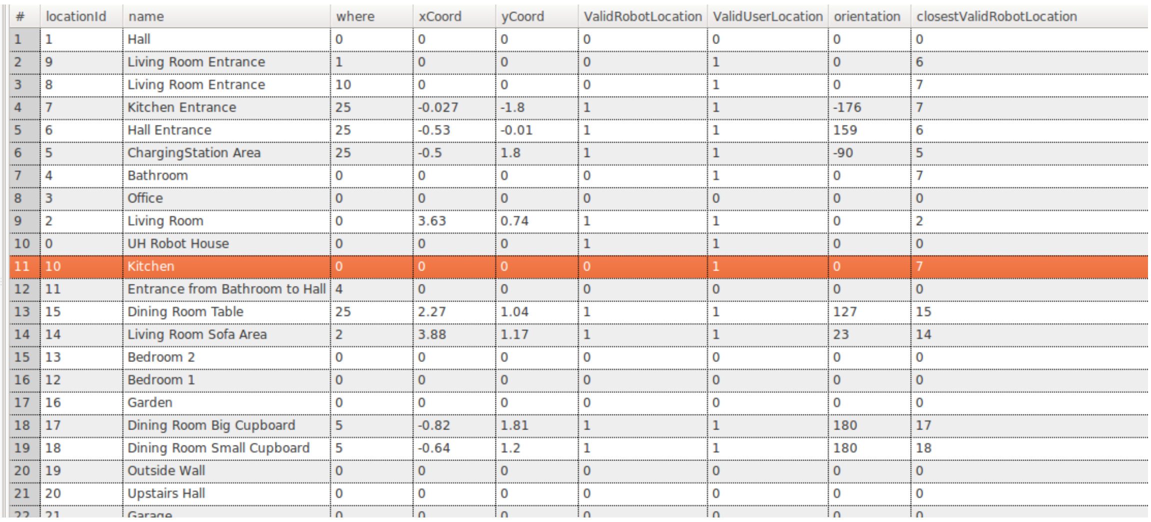
(c)

Figure 13 - Tables in the Accompany database that store users' exceptional cases proxemics preferences data needed by Exceptional Cases Proxemics Preferences Based Algorithm.

Location Ontology Based Proxemics Algorithm

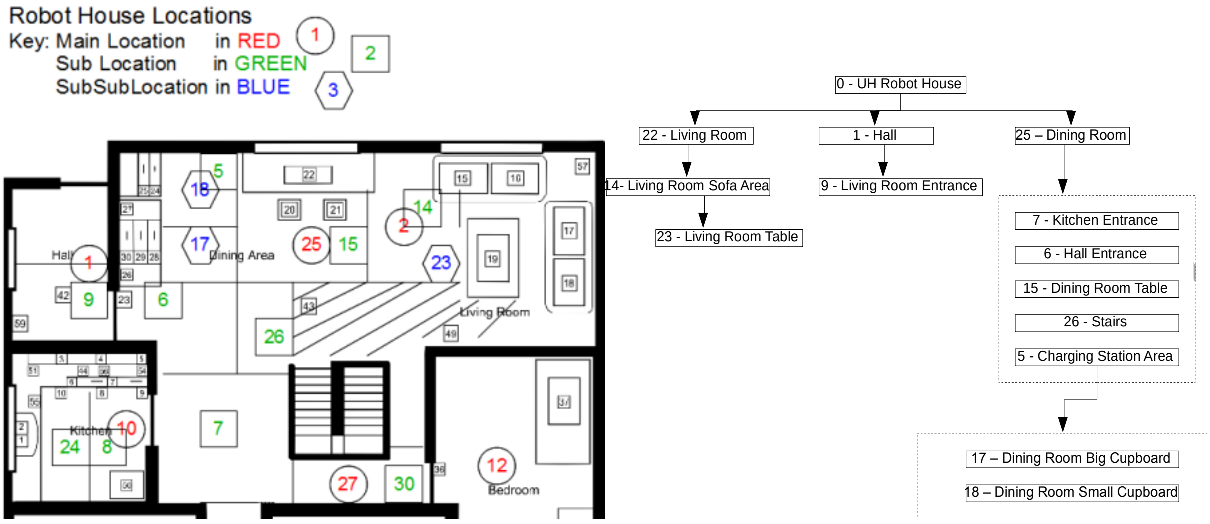
The Location Ontology Based Proxemics algorithm deals with cases where it is not possible for the robot to approach the user at their location. This can be because there is no valid path for the robot to approach the user (e.g. the user is behind a doorway that is too small for the robot to go through), or the robot cannot go into a small confined area such as a kitchen. In these situations, it is not feasible to use the General Proxemics Based Algorithm nor Exceptional Cases Based Proxemics Algorithm unless an exceptional case proxemics was previously set by the participant for this location.

The algorithm uses the user's location information to search for a location closest to the user's current location that is accessible by the robot in the Location table shown in Figure 14a. The Location table was created based on the UH Robot House location ontology diagram shown in Figure 14b, which was created by mapping the UH Robot House map with the Hierarchical diagram of the locations ontology shown in Figure 14c.



#	locationId	name	where	xCoord	yCoord	ValidRobotLocation	ValidUserLocation	orientation	closestValidRobotLocation
1	1	Hall	0	0	0	0	0	0	0
2	9	Living Room Entrance	1	0	0	0	1	0	6
3	8	Living Room Entrance	10	0	0	0	1	0	7
4	7	Kitchen Entrance	25	-0.027	-1.8	1	1	-176	7
5	6	Hall Entrance	25	-0.53	-0.01	1	1	159	6
6	5	ChargingStation Area	25	-0.5	1.8	1	1	-90	5
7	4	Bathroom	0	0	0	0	1	0	7
8	3	Office	0	0	0	0	0	0	0
9	2	Living Room	0	3.63	0.74	1	1	0	2
10	0	UH Robot House	0	0	0	1	1	0	0
11	10	Kitchen	0	0	0	0	1	0	7
12	11	Entrance from Bathroom to Hall	4	0	0	0	0	0	0
13	15	Dining Room Table	25	2.27	1.04	1	1	127	15
14	14	Living Room Sofa Area	2	3.88	1.17	1	1	23	14
15	13	Bedroom 2	0	0	0	0	0	0	0
16	12	Bedroom 1	0	0	0	0	0	0	0
17	16	Garden	0	0	0	0	0	0	0
18	17	Dining Room Big Cupboard	5	-0.82	1.81	1	1	180	17
19	18	Dining Room Small Cupboard	5	-0.64	1.2	1	1	180	18
20	19	Outside Wall	0	0	0	0	0	0	0
21	20	Upstairs Hall	0	0	0	0	0	0	0
22	21	Garage	0	0	0	0	0	0	0

(a)



(b) (c)

Figure 14 - UH Robot House location ontology. a) Example mapping of location ontology in the Location table in the Accompany database, b) visual mapping of location ontology on UH Robot House Map, and c) Hierarchical diagram of the location ontology.

Example scenario of the Care-O-bot utilising the context-aware planner for approaching a user.

The client calls the Context-Aware Planner's Proxemics service to obtain suitable target positions for the robot to approach the user. To do this, the client needs to provide the following six parameters: the user's id, posture, position and orientation (x, y, theta) in map coordinate frame, and the robot's task at the target positions.

Utilising the information provided, the planner retrieves the user's location (*locationId*) from the database. The user's *locationId* is automatically updated by an external module "UHCore/location.py".

The *locationIds* provide a link to the symbolic names of different location in the map (i.e. kitchen, living room etc.) that provides useful contextual information. The user's *locationId* is updated using the information obtained from the omni-directional camera based person tracking module developed as part of WP4.

First the planner goes through **SensorBasedProxemicsPreferences** table to search for sensors that are triggered at the user’s location. By obtaining the *exceptionCasesProxemicsId* associated with the sensor that is triggered, it can then retrieve the target coordinates for the *exceptionCasesProxemicsId* from the **ExceptionCaseProxemicsPose** table and send it to the robot.

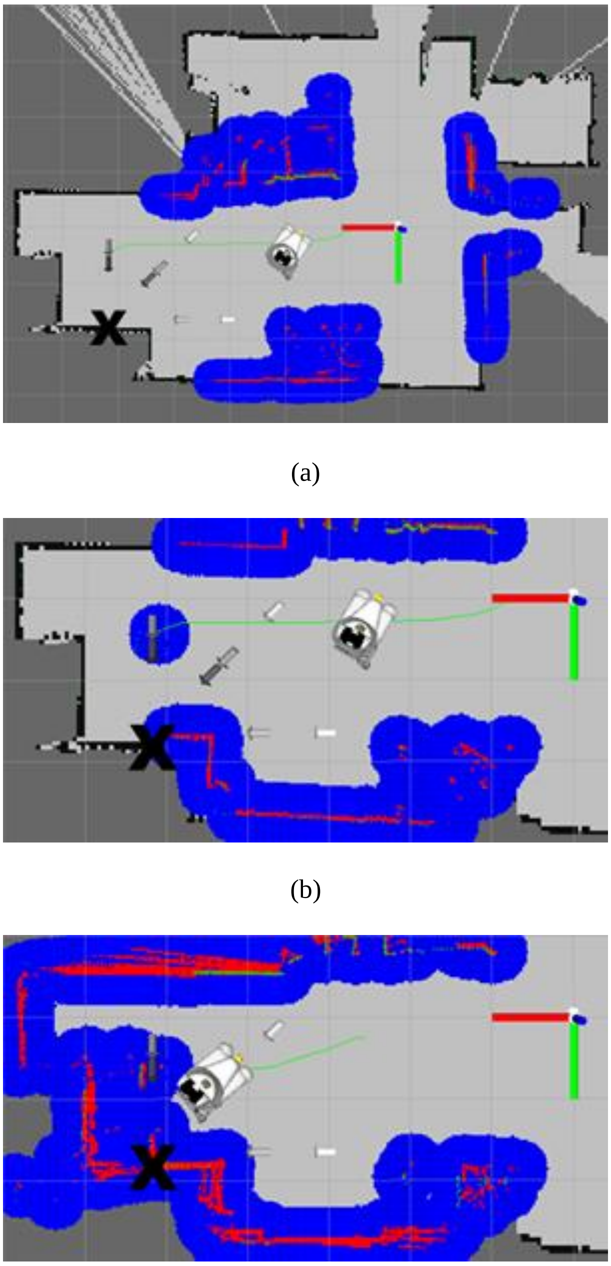
If the planner fails to find any sensors triggered around the user’s location, it will search the **LocationBasedProxemicsPreferences** table for entries at the user’s current location. If the algorithm manages to find a suitable entry in that table, it will retrieve the target coordinate for the *exceptionCasesProxemicsId* that is associated with that entry from the **ExceptionCasesProxemicsPose** table and send it to the robot.

If the planner fails to find the user’s location entry in the **LocationBasedProxemicsPreferences** table, it will then use the **Locations** table in the database to check if the user’s location is reachable by the robot. If the user’s location is not reachable by the robot, the algorithm will search for the closest reachable location using the **LocationOntology** information and send the coordinates of that location to the robot. Note that the coordinates for all the locations in the **Locations** table are selected for practical reasons in order to ensure the robot is able to reach the user for interaction.

If the user’s location is reachable by the robot, the algorithm will retrieve the user’s proxemics preferences from the **UserProxemicsPreferences** table to create a prioritised list of all the possible robot positions and orientations around the user.

By utilising the static map, the algorithm then eliminates all the possible robot positions that cannot be occupied by the robot (i.e. obstacle or too close to obstacle), cannot be reached by the robot (i.e. due to obstacle or small passage) and finally, it eliminates positions that are not in the same location as the user.

The possible target coordinates that pass the elimination processes are then sent back to the client in a prioritised arrangement (see Figure 15a). The robot can then navigate to the first coordinate on the list. If the first position is blocked by a dynamic obstacle (see Figure 15b), the robot can select the next coordinate down the prioritised list until it is able to reach one of the coordinates (see Figure 15c).



(a)

(b)

(c)

Figure 15 - Evaluation scenario where the Care-O-bot approaches an experienced user (with a preference for the robot to approach from the left side) for a fetch and carry task. The user is sitting in the living room sofa location A (marked by X). The detected obstacles are shown as red dots while the expanded obstacles (the "forbidden zones") for the robot are marked with blue dots.

Technical Evaluation of Context Aware Planner

A technical evaluation (formative study) was conducted to examine the efficacy of the context-aware planner. The technical evaluation focused on how well the context-aware planner would perform and adapt to a user's preferences within the constraints of a domestic environment. We identified three locations of interest, the Kitchen, Dining room and Living room. These were locations where the Care-O-bot® would be likely to approach the user for interaction. Figure 16 shows a diagram of the UH Robot House with the three locations of interest highlighted.

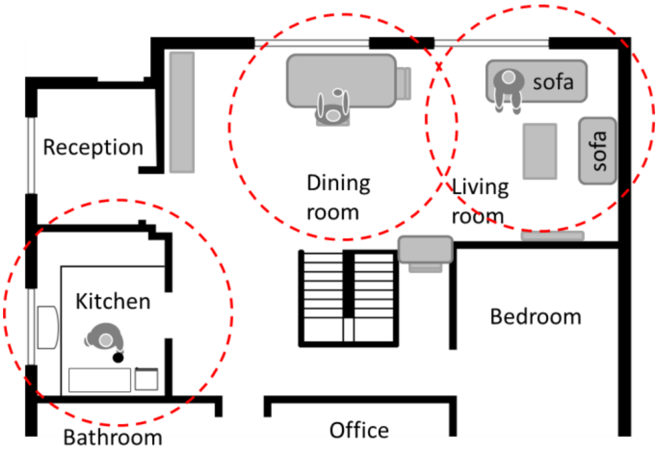
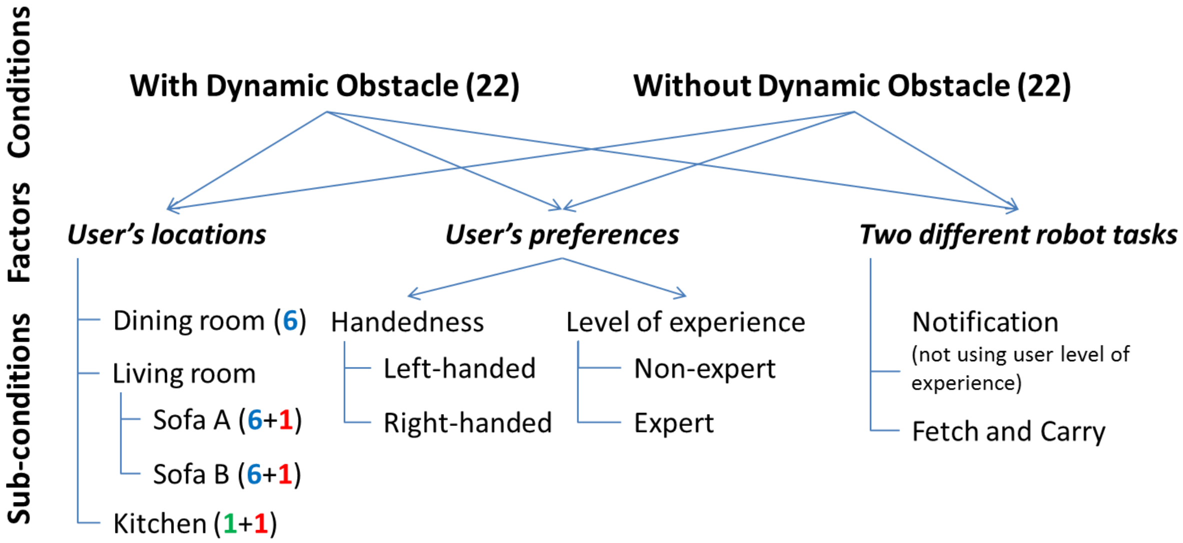


Figure 16 - The 3 main locations used for the technical evaluation are locations that the Care-O-bot would be likely to approach the user for interaction.

It is important that the planner can cope with different users, taking into account their interaction experience with robots and proxemics preferences, in order to ensure that the planner provides target coordinates that are appropriate for both the user and the task the robot is going to perform at the user's location regardless of the presence of dynamic obstacles at that location or whether the user is located at a small confined space such as in the UH Robot House Kitchen.

Figure 17 shows the evaluation conditions diagram that was developed to cover all the conditions highlighted above. There are two main conditions, the first condition looks at a situation when there are no unexpected obstacles in the environment while the second condition looks at a situation when there are unexpected obstacles in the environment. As shown, within each condition, there are three different factors: *Location*, *User*, and *Robot task*.



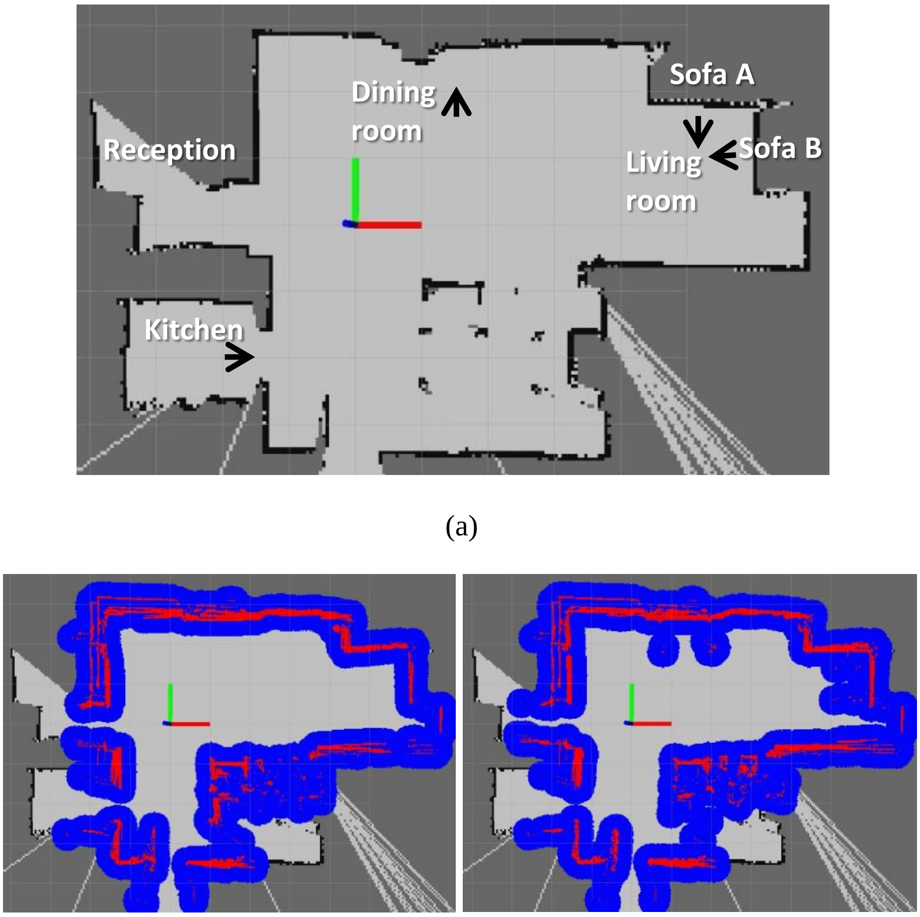
```
//General Proxemics Preferences Based algorithm
4 = Dining room (1) * Handedness (2) * Level of Experience (2) * Fetch and Carry (1)
2 = Dining room (1) * Handedness (2) * Notification (1)
8 = Living room (2) * Handedness (2) * Level of Experience (2) * Fetch and Carry (1)
4 = Living room (2) * Handedness (2) * Notification (1)
//Exceptional Cases for Sensor based proxemics algorithm
2 = Living room (2) * Notification (1)
//Exceptional Cases for Location based algorithm)
1 = Kitchen * Notification (1)
//Location Ontology algorithm
1 = Kitchen * Notification (1)
```

Figure 17 - Evaluation conditions diagram and its associated legend to illustrate all the experimental conditions used in the evaluation.

The Location factor looks at the planner adapting to different environment configurations in a domestic environment, in this case the Kitchen, the Dining room and the Living room (Sofa location A and Sofa location B). The Users factor looks at the planner adapting to different user preferences (i.e. right handed approach or left handed approach) and different interaction experience with robots. The Robot tasks looks at the planner adapting to different tasks carried out by the robot during interaction (i.e. Notification or Fetch and Carry tasks).

Overall for each condition we have 22 different configurations per condition for the technical evaluation. This consists of 6 configurations for the Dining room, 14 configurations for the Living room and 2 configurations for the Kitchen. In total we have 44 different configurations for both conditions. During the experiment, we conducted 3 trials for each configuration for consistency purposes. This resulted in a total of 132 trials. The trials were conducted over a period of four days. The first two days involved all the trials for the environment with *no dynamic obstacles condition*. The second two days involved all the trials for the environment *with dynamic obstacles condition*.

Figure 18a shows the UH Robot House Map used by the navigation system. The locations within the map are labelled for easy reference to Figure 16. Note that the user's locations used for the experiment are indicated by arrow head. Figure 18b and 18c shows the Robot House map from the perspective of the navigation system where detected obstacles (i.e. red dots) and the expanded obstacles (i.e. blue areas) are added to indicate the forbidden zones for the robot. Figure 18b shows the map for the no dynamic obstacle condition while Figure18c shows the map where dynamic obstacle were added to the environment, specifically the dining table at the Dining room and a coffee table in the Living room.



(a)

(b) (c)

Figure 18 - (a) Overview of the robot house map with location labels, (b) overview of the robot "forbidden zones", and (c) overview of the robot "forbiden zones" with the presence of dynamic obstacles.

Two virtual users were created in the ACCOMPANY database for the experiment. One of the users had a preference for the robot to approach from the right hand side, while the other user had a preference for the robot to approach from the left.

For the experiment, the robot always started its approach to the user from its home position indicated by the coordinate frame shown in figure 18.

The experiment involved an experimenter and an actor. The actor’s job was to act as the user to sit at one of the locations shown in Figure 18a during the experiment. The user’s sitting location varied depending on the configurations of the specific trial being conducted.

The experimenter used the Context-Aware Planner standalone client from a terminal to call to the proxemics service by providing the necessary parameters. Upon receiving the ranked target coordinates from the planner, the standalone client then issued these coordinates to the Care-O-bot® as described previously.

During the experiment, a video camera was setup to record the robot's behaviours. Screen capture tools were also used to capture the outputs from the Navigation System, the Context Aware Planner, the standalone client as well as the visual display from ROS 3D visualization tool – rviz. This allowed us to collect all the data necessary for improving the system.

Results

Overall, the results show that the Care-O-bot® successfully approached the user in all of the 132 trials conducted to evaluate the performance of the Proxemics system coping with 42 different configurations (3 trials each) in the robot house. The results shown in Figure 19 indicate that on average, the Context-aware proxemics planner took less than 30ms to provide ranked target coordinates in response to the standalone proxemics client request.

The experiment also revealed that the idea of the context-aware planner providing ranked target coordinates is useful to ensure that the navigation system has other options to reach the user in the cases of unexpected situations as encountered during the experiment. There were a total of 30 occurrences where the robot could not reach the first target coordinates (position or orientation) due to phantom obstacles detected by the laser scanner or due to inaccurate localisation. However, in 24 of the occurrences, the robot was able to reach the user with second target coordinates, in two of the occurrences the robot was able to reach the user using the third target coordinates, in three of the occurrences the robot was able to reach the user using the fourth target coordinates, and in one occurrence the robot was able to successfully reach the user using the fifth target coordinates.

During the experiment we also discovered 6 occurrences where the ROS move_base action server, which is responsible for taking a given goal in the world and then attempt to reach it with a mobile base, was not ready to receive the target pose. In those situations, the standalone client had to wait from a minimum of 5 second to a maximum of 30 second for the move_base action server to be ready.

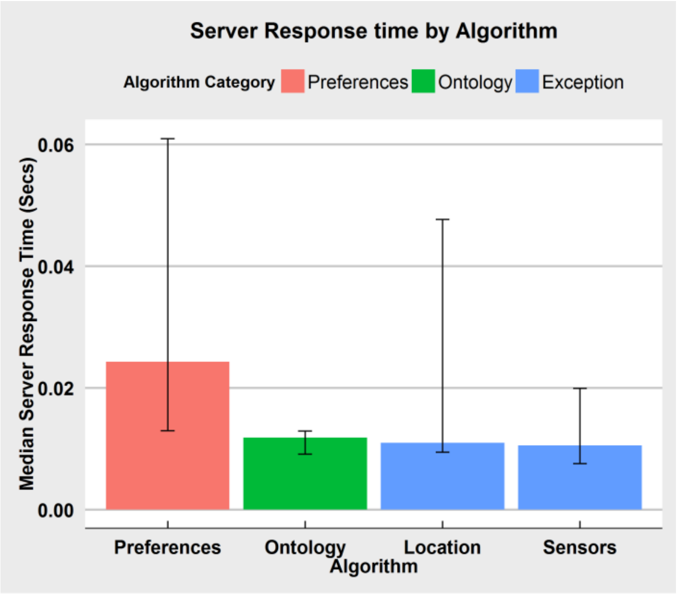


Figure 19 - Response time performance of the context-aware proxemics planner server.

6. Implementation and Integration of Empathic Behaviours

This section describes the implementation and integration of the concept of perceptual crossing described in D2.2, in particular the “I See You Seeing Me” and the “Walk with Me” empathic behaviours.

I See You Seeing Me.

The implementation of “I See You Seeing Me” scenario is based on the concept of perceptual crossing described in D2.2. Here we integrate the “I See You Seeing Me” scenario as a tool for the robot initiate interaction by expressing its presence to the user and to make the user feel that they are being seen by the robot, who is aware of their presence.

To achieve this, we utilise proxemics theory to help the robot decide when to engage and disengage with the user. The decision was that the robot should only engage with the user when the user entered the robot’s social space (i.e. 3.7m). Also for safety reasons the robot should stop tracking the user when they have entered its intimate space (i.e. 45cm).

We also utilise both the UvA omni-directional camera based person tracking system (see WP4) and the Fraunhofer IPA laser based person tracking system to give increased confidence in person tracking. For example, the UvA vision based system has the potential to be unable to confound the robot and the user locations in situations where both the user and the robot are moving in very close proximity of each other. The Fraunhofer IPA laser based system on the other hand is very reliable when the user is in close proximity to the robot, but it suffers from false detection when in the presence of leg-like features such as the combination of table and chair legs in the dining area. A few trial and error sessions were conducted with the Fraunhofer IPA laser based system to determine the range of detection that would give the fewest false positives and it was found that in the UH Robot House, the system performed best when the target was less than 1.5m from the robot. Therefore it was decided that the target detected by the laser based system should have a higher priority than the target detected by the UvA system when both targets are within 1.5m from the robot. The system has to rely solely on the UvA system for targets that are between 1.5m to 3.7m.This decision allows us to overcome the weaknesses of both tracking systems and to obtain more reliable human tracking data needed for integrating the I See You Seeing Me scenario.

Figure 20 shows an overview of how I See You Seeing Me implementation with the Care-O-bot® for the ACCOMPANY System. The module’s main processing loop runs at a rate of 100Hz. The data from both trackers is independently received and processed through ROS callback mechanisms which get triggered at the point of data arrival.

In a processing loop, the module will select the closest detected valid target, i.e. that are between 0.5m to 3.7m from the robot, and then check the ACCOMPANY database to see if the robot control resources were available (from the COBScheduler robot control system). Only when the robot control resources are available, will the system activate the I See You Seeing Me behaviour. The module then looks at the bearing of the target, and if it is more than 5 degree, it will turn its torso towards the user. If the target bearing is more than 18 degree, the robot will also start rotating its base until its base is facing the user. The parameters of 5 degree and 18 degree thresholds were obtained through trial and error, aiming to allow the robot to give the impression of being aware and interested in engaging in an interaction with the user. In the event that a user stands in front of the robot for more than 2 seconds, the module will set the Robot Initiate Interaction flag to true. This flag can be used by the COBCoreScheduler to initiate face recognition and interact with the user.

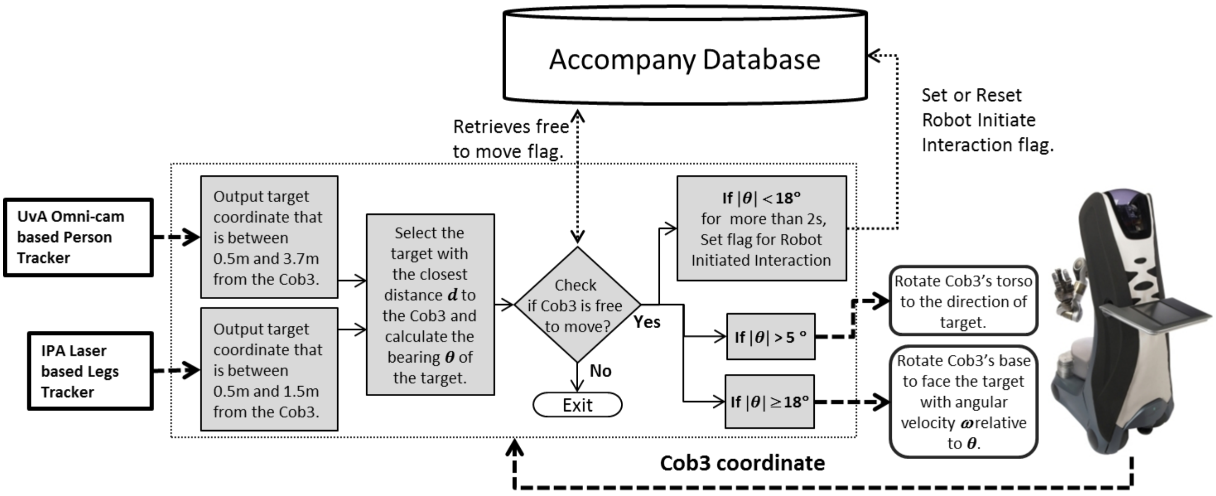

Accompany Database
Retrieves free to move flag.
Set or Reset Robot Initiate Interaction flag.
UvA Omni-cam based Person Tracker
IPA Laser based Legs Tracker
Output target coordinate that is between 0.5m and 3.7m from the Cob3.
Output target coordinate that is between 0.5m and 1.5m from the Cob3.
Select the target with the closest distance d to the Cob3 and calculate the bearing θ of the target.
Check if Cob3 is free to move?
Yes
No
Exit
If |θ| < 18° for more than 2s, Set flag for Robot Initiated Interaction
If |θ| > 5 °
If |θ| ≥ 18°
Rotate Cob3's torso to the direction of target.
Rotate Cob3's base to face the target with angular velocity ω relative to θ.
Cob3 coordinate

Figure 20 - An overview of the empathic behaviour algorithm for I See You Seeing Me.

Walk with Me

In order to let the robot accompany a person while walking (see Figure 21), a walk together function has been implemented by ACCOMPANY partner Fraunhofer IPA. This module receives localization information on the user from either the ceiling camera based tracking system or from a leg tracking algorithm based on the laser scanner sensors mounted directly on the robot. The user position information comes with a speed vector representing recent movement. Using both user location and speed it is straightforward to compute two desired robot poses at the left and right side of the user. The one which is closest to the current robot position is chosen as the next navigation goal for the robot. This way the robot never has to travel a long way to stay at either side of the user. For a detailed explanation of the algorithm please see deliverable D5.5, Section 2.2.6.

The walk together function is implemented in Python as a state machine, which can be simply accessed from the robot behaviour scheduler (COBCoreScheduler) through an extension for executing arbitrary behaviour scripts. This extension was introduced by Fraunhofer IPA to provide easy access to complex robot behaviours beyond the scope of standard script server directives (see also D5.5, Section 2.2.3). The walk together script finishes upon a termination criterion, which can be specified by a parameter on execution. Usually, this criterion is linked to the arrival at a certain area, e.g. the kitchen or the door.

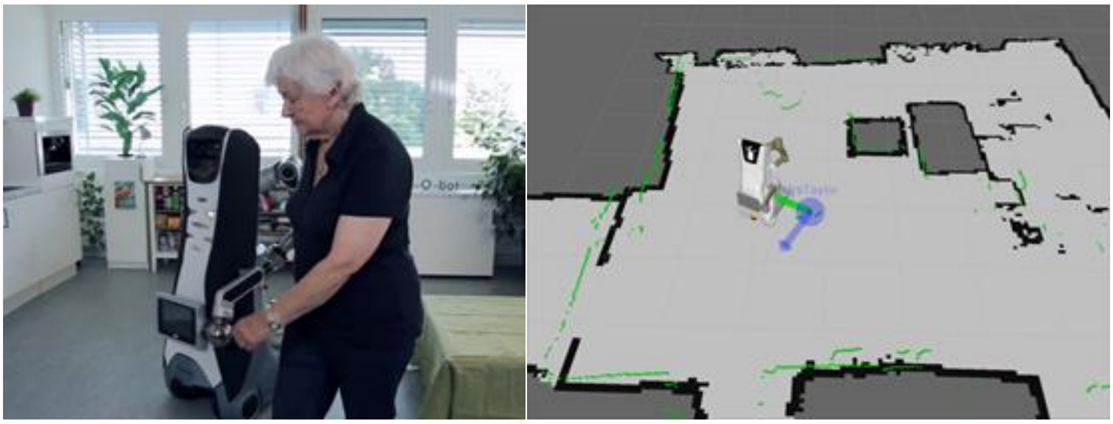


Figure 21 – Left) The Care-O-bot exhibiting Walk with Me behaviour to accompany a user, Right), The Walk with Me algorithm uses the person's speed vector (blue arrow), obtained from UvA omni-camera person tracker, to compute the desired robot location (green arrow).

7. Discussion and Future Work

This deliverable has reported the work done for T2.4. This work draws on both the robot task requirements listed in D1.2 and the scenario reported in D1.3 to formulate the requirements for the context-aware planner and the development of a knowledge-driven activity recognition system to provide contextual information of the user that is necessary for the context-aware planner.

Two different user studies were conducted in phase two highlighting the importance of robot etiquettes in domestic environment as well as the effect of contingent behaviour on robot appearance. The main findings from these studies indicate that there are differences between users' preferences with regards to robot behaviours and proxemics, and that these do change with the context of the interaction.

The developed context-aware planner takes these into account to allow the user to personalise their proxemics preferences. This lets the robot cope with specific user's preferences across different contexts.

Results from a formative evaluation have shown that the developed context-aware planner successfully and reliably utilised the user preferences and contextual information to provide suitable, socially acceptable ranked target coordinates for the Care-O-bot® to approach the user.

Implementation of the empathic behaviours "I See You Seeing Me" and "Walk with Me" was done in conjunction with exploratory user studies exploring the importance of robot contingent behaviour. This would allow the robot to utilise perceptual crossing to initiate interaction and accompany the user in an acceptable manner.

Future work include instantiating the context aware planner and the empathic behaviours robustly within a scenario suitable for evaluating these behaviours in a natural setting in context.

8. References

G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton (1997). ***Cyberguide: a mobile context-aware tour guide**.* Wireless Networks, 3(5) pp.421-433.

ACCOMPANY D1.1 (2012). ***Status of elderly care provision in Europe, potential for service robotics.***

ACCOMPANY D1.2 (2012). ***Report on user and system requirements and first outline of system functionality***.

ACCOMPANY D1.3 (2012). ***Phase one scenarios and report on system functionality.***

ACCOMPANY D2.2 (2013). ***Low-fidelity prototypes and mock-ups for perceptual crossing.***

ACCOMPANY D3.3 (2013). ***First design and technical implementation of computational memory architecture.***

ACCOMPANY D3.4 (2014). **Final evaluation of ACCOMPANY computation memory architecture.**

ACCOMPANY D4.4 (2013). ***Data fusion and activity recognition in household chores****.*

ACCOMPANY D5.5 (2014). ***Report on showcase activities****.*

M. Baldauf, S. Dustdar and F. Rosenberg (2007). ***A survey on context-aware systems***. International Journal of Ad Hoc and Ubiquitous Computing, 2(4), pp.263-277.

Ling Bao and Stephen Intille. ***Activity recognition from userannotated acceleration data***. In Alois Ferscha and Friedemann Mattern, editors, Pervasive Computing, volume 3001 of Lecture Notes in Computer Science, pages 1–17. Springer Berlin / Heidelberg, 2004.

J. K. Burgoon, and J. B. Walther (1990). **Nonverbal expectancies and the evaluative consequences of violations**. Human Communication Research,17(2), 232-265.

K. Cheverst, N.Davies, K. Mitchell, A. Friday and C. Efs-tratiou (2000). **Developing a context-aware electrocnic tourist guide: some issues and experiences**. Proceeding of the SIGCHI conference on Human Factors in Computing Systems, ACM Oress, New York, usa, pp.17-24.

James L. Crowley, Joëlle Coutaz, Gaeten Rey, and Patrick Reignier (2002). ***Perceptual Components for Context Aware Computing***. In Proceedings of the 4th international conference on Ubiquitous Computing (UbiComp '02), Gaetano Borriello and Lars Erik Holmquist (Eds.). Springer-Verlag, London, UK, UK, 117-134.

Liming Chen, C.D. Nugent, and Hui Wang (2012). ***A Knowledge-Driven Approach to Activity Recognition in Smart Homes***. IEEE Transactions on Knowledge and Data Engineering, 24(6), pp. 961-974.

A. K. Dey, M. Futakawa, D. Salber and G. D. Abowd (1999). ***The Conference Assistant: Combining Context-Awareness with Wearable Computing***. Proceedings of the 3rd International Symposium on Wearable Computers, San Francisco, CA. pp.21-28.

A. K. Dey, D. Salber and G. D. Abowd (2001). ***A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications***. Human-Computer Interaction Journal 16(2-4), pp. 96-166, 2001.

I. Duque, K. Dautenhahn, K.L. Koay, I. Willcock and B. Christianson (2012). ***Knowledge-driven User Activity Recognition for a Smart House. Development and Validation of a Generic and Low-Cost, Resource-Efficient System***. Submitted for review.

GEO (Green Energy Options): http://www.greenenergyoptions.co.uk/. Last accessed October 2012.

R. M. Gustavsen (2002). ***Condor – an application framework for mobility-based context-aware applications***. Proceedings of the Workshpo on Concepts and Models for Ubiquitous Computing, Goeteborg, Sweden.

E. T. Hall (1966). ***The Hidden Dimension***. Garden City, N.Y.: Doubleday.

T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger and J. Altmann (2002). ***Context-awareness on mobile devices - the hydrogen approach***. Proceedings of the 36th Annual Hawaii International Conference on System Sciences, pp.292-302.

Ninghang Hu, Gwenn Englebienne, Ben Kröse (2014). ***A Two-layered Approach to Recognize High-level Human Activities*****.** In Proc. of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2014

K.L. Koay, K. Dautenhahn, S.N. Woods and M.L. Walters (2006). ***Empirical Results from Using a Comfort Level Device in Human-Robot Interaction Studies***. in Proceeding of 1st Annual Conference on Human-Robot Interaction (Salt Lake City, Utah, March 2006), 194-201.

Kheng Lee Koay , Dag Sverre Syrdal , Michael L. Walters , and Kerstin Dautenhahn (2007). ***Living with Robots: Investigating the Habituation Effect in Participants? Preferences during a Longitudinal Human-Robot Interaction***. IEEE RO-MAN (Jeju Island, Korea, 26-29 August 2007), 564-569.

Kheng Lee Koay, Dag Sverre Syrdal, Michael L. Walters and Kerstin Dautenhahn (2009). ***Five Weeks in the Robot House - Exploratory Human-Robot Interaction Trials in a Domestic Setting***. Proc. of the Second International Conferences on Advances in Computer-Human Interactions. February 1-6, 2009 - Cancun, Mexico.

Kheng Lee Koay, Michael L. Walters, Alex May, Anna Dumitriu, Bruce Christianson, Nathan Burke and Kerstin Dautenhahn (2013). **Exploring Robot Etiquette: Refining a HRI Home Companion Scenario Based on Feedback from Two Artists Who Lived with Robots in the UH Robot House.** Proc. of the 5th International Conference for Social Robotics, October 2013, Bristol, pp. 290-300.

S. K. Mostefaoui and B. Hirsbrunner (2003). ***Towards a Context-Based Service Composition Framework***. Proceedings of the International Conference on Web Services, Las Vegas, Nevada, pp.42-45.

C. Palmisano, A. Tuzhilin, M. Gorgoglione (2008). ***Using Context to Improve Predictive Modeling of Customers in Personalization Applications***. IEEE Transactions on Lnowledge and Data Engineering 20(11), pp.1535-1549.

Kasteren, T.L.M. van, Englebienne,G. and & Kröse, B.J.A (2010). ***Activity recognition using semi-markov models on real world smart home datasets***. J. Ambient Intell. Smart Environ.,2(3):311-325, 2010.

Tim van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse (2008). ***Accurate activity recognition in a home setting***. In Proceedings of the 10th international conference on Ubiquitous computing, UbiComp’08, pages 1–9, New York, NY, USA, 2008.

P. Korpipää, J. Mäntyjärvi (2003). ***An Ontology for Mobile Device Sensor-Based Context Awareness***. Proceeding of CONTEXT vol 2680 of Lecture Notes in Computer Science, pp.451-458

P. Prekop and M. Brunett (2003). ***Activities, context and ubiquitous computing***. Special Issue on Ubiquitous Computing Computer Communications, 26(11), pp.1168-1176.

Joan Saez-Pons, Hagen Lehmann, Dag Sverre Syrdal, Kerstin Dautenhahn. ***Development of the Sociability of Non-Anthropomorphic Robot Home Companions***. 4th IEEE International Conference on Development and Learning and on Epigenetic Robotics(ICDL-EpiRob 2014). Genoa, Italy. October 13th-16th, 2014.

D. Sanchez, M. Tentori, and J. Favela. **Activity recognition for the smart hospital**. Intelligent Systems, IEEE, 23(2):50 –57, March 2008.

A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, W. Van de Velde (1999). **Advanced interaction in context**. Proceedings of Fisrst International Symposium on Handheld and Ibiquitous Computing, Karlsruhe, Germani, pp.89-101.

A. Schmidt, M. Beigl and Hans-w. Gellersen. **There is more to Context than Location**. Computers and Graphics,1998, 23, pp. 893--901

SMACH. http://www.ros.org/wiki/smach

Y. Sumi, T. Etani, S. Fels, N. Simonet, K. Kobayashi and K. Mase (1998). ***C-map: Building a context-aware mobile assistant for exhibition tours***. Community Computing and Support Systems, Social Interaction in Networked Communities. Springer-Verlag, London, UK, pp.137-154.

Leila Takayama, Caroline Pantofaru (2009). Influences on proxemic behaviors in human-robot interaction, Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems, pp.5495-5502, October 10-15, 2009, St. Louis, MO, USA.

Emmanuel Tapia, Stephen Intille, and Kent Larson. ***Activity recognition in the home using simple and ubiquitous sensors***. In Alois Ferscha and Friedemann Mattern, editors, Pervasive Computing, volume 3001 of Lecture Notes in Computer Science, pages 158–175. Springer Berlin / Heidelberg, 2004.

M.L. Walters, K. Dautenhahn, K.L. Koay, C. Kaouri, R. te Boekhorst, C.L. Nehaniv, I. Werry and D. Lee (2005). ***Close encounters: Spatial distances between people and a robot of mechanistic appearance***. in Proceedings of 5th IEEE-RAS International Conference on Humanoid Robots (Tsukuba, Japan, December 2005), 450-455.

M. L. Walters, K. Dautenhahn, S. N. Woods, K. L. Koay, R. te Boekhorst, D. Lee (2006**) Exploratory Studies on Social Spaces between Humans and a Mechanical-looking Robot**. Connection Science 18(4), pp. 429-442.

Michael L. Walters, Kerstin Dautenhahn, Sarah N. Woods, Kheng Lee Koay (2007). **Robotic Etiquette: Results from User Studies Involving a Fetch and Carry Task**. in Proceeding of 2nd Annual Conference on Human-Robot Interaction (March 8-11, 2007, Arlington, Virginia, USA) pp. 317-324.

M. L. Walters, K. Dautenhahn, R. te Boekhorst & K. L. Koay (2009) **An Empirical Framework for Human Robot Proximity, Proceedings New Frontiers in Human-Robot Interaction**, a symposium at the AISB2009 Convention, Heriot Watt University, 8-9 April 2009, Edinburgh, Scotland, published by SSAISB, ISBN - 190295680X, pp. 144-149

R. Want, A. Hopper, V. Falcao, and J.Gibbons (1992). ***The active badge location system***. ACM Transactions on Information Systems, 10(1), pp.91-102.

M. Weiser (1991). ***The Computer for the 21st Century***, Scientific American pp.94-104.

T. Winograd (2001). ***Architectures for context***. Human-Computer Interaction (HCI) Journal, 16(2), pp. 401-419.

Zigbee: http://www.zigbee.org/. Last accessed October 2012.