Artificial Intelligence Applied to Assistive Technologies and Smart Environments: Papers from the 2015 AAAI Workshop

The Implementation of a Planning and Scheduling Architecture for Multiple Robots Assisting Multiple Users in a Retirement Home Setting

Tiago Vaquero, Sharaf Mohamed, Goldie Nejat, J. Christopher Beck

Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada M5S 3G8
{tvaquero, nejat, jcb}@mie.utoronto.ca, sharaf.mohamed@mail.utoronto.ca

Abstract

Our research focuses on the use of Planning & Scheduling (P&S) technology for a team of robots providing daily assistance to multiple elder adults living in retirement facilities. Multi-user assistance and group-based activities require robots to plan and schedule their human-robot interaction (HRI) activities based on the specific needs, time constraints, availability and preferences of the multiple users. In this paper, we introduce and implement a novel centralized system architecture that can manage real P&S scenarios with multiple socially assistive robots, multiple users and their individual schedules, and single- and multi-person assistive activities. We describe how the main components of the proposed P&S architecture are integrated to control the robots, and to generate and monitor sequences of temporally annotated activities using off-the-shelf temporal planners. We verify that the architecture can manage realistic scenarios with three assistive robots, twenty users, and several single- and group-based activity requests during a single day.

Introduction

The rapid growth of the world's elderly population and the current shortage of healthcare professionals have led to increasing efforts to develop robotic systems that can be used to assist the elderly in eldercare environments (e.g. Pineau et al. 2003; McColl et al. 2013). Some of these robots have incorporated automated planning and scheduling (P&S) systems to enhance the assistive behaviors of the robots (Pineau et al. 2003; Pollack 2005; Cesta et al. 2011). In general, the majority of this research has focused on human-robot interaction (HRI) scenarios with a single robot assisting a single user.

To-date, only a handful of works have considered scenarios where a robot interacts with multiple people at the same time, e.g. (Montemerlo et al. 2002; Bennewitz et al. 2005; Petrick and Foster 2013). However, these robots do not actively distinguish between users to provide personalized interactions during multi-user activities. With respect to reasoning about multiple users, the Cobot robots (Coltin et al. 2011) have been designed to plan and schedule HRI activities, including semi-autonomous telepresence and office tasks, based on requests from several users. Yet the planning and scheduling are managed independently and the user schedules are not considered as constraints for the robots' activities. Given the variety of users' needs and availabilities, robot facilitated multi-user assistive activities require robots to plan, schedule, and customize their HRI interactions to the needs, time constraints, availability and preferences of each individual during the day.

An environment which consists of multiple robots assisting multiple users with single- and multi-person HRI activities based on the schedules of the users has not yet been addressed in both the robotics and P&S literature. In this work we address a robotics-based problem in which an integration of both planning and scheduling is required. Namely, the long-term objective of our work is to deploy a team of mobile socially assistive robots in retirement homes to engage residents in stimulating recreational activities on a daily basis (Louie et al. 2013; Louie et al. 2014). In this paper, we propose the development and implementation of a unique centralized P&S system architecture to plan and schedule the daily assistive activities of a group of socially assistive robots to engage with multiple residents living in a retirement home based on the residents' schedules. We focus on two representative stimulating activities: *telepresence* (single-person activity) and *Bingo* (multi-person activity). For a telepresence session, an assigned robot autonomously navigates to the user in his/her private room, prompts him/her for a video call, starts the call, tracks the user during the session and ends the call. For the Bingo game, robots autonomously find and remind participants of the game prior to its start, then an assigned robot navigates to a specified location and acts as the game facilitator, calling out numbers, verifying Bingo cards, prompting players to mark missed numbers and celebrating with winners. Since these activities can deplete the robot's batteries, a recharging activity may also be necessary.

The proposed system architecture addresses a realistic and complex combination of reasoning about activity, resources (e.g., the robots), time windows (e.g., user availability), temporal constraints (e.g., activity deadlines), and metric quantities (e.g., battery level). There has been a recurring discussion in the literature regarding the challenges

---

of combining these elements, which are often investigated independently (Smith et al. 2000). However, developing solvers for P&S applications with all these features is still an open challenge. The novelty of our work is based on: 1) the design of a new realistic problem for the use of assistive robotics and P&S technology in retirement home settings which considers multiple robots, multiple users and their schedules, and single-user and multi-user socially and cognitively stimulating activities, and 2) the development and implementation of a centralized system architecture using off-the-shelf temporal planners to solve the proposed problem, and plan and schedule the robots' assistive activities in the target human-centered environment.

The Problem

Herein, we define the main elements of the proposed realistic problem: the environment in which the residents (users) and robots interact, the constraints, and the overall goal. We have adapted the problem we first introduced in (Vaquero et al. 2014) for real world scenarios.

The Retirement Home Environment: We consider a floor within a retirement home which consists of rooms, corridors and hallways that are discretized as a set of locations, L (l_1 ... l_n), within which the users and robots will interact. The set of locations and the distance between any two locations ($d_{i,j}$) are known a priori.

Users: The users are residents of the retirement home. We consider a set of users, U (u_1 ... u_n), in which each user u_k has his/her own *profile*. The profile consists of the user's private *room* location; his/her interest to participate in Bingo games and telepresence sessions; and his/her own distinct *schedule* for the day, representing the user availability (in time and space) for interaction with a robot.

A typical day is from 7am-7pm. All users are considered unavailable during meal times, i.e. breakfast (8-9am), lunch (12-1pm), and dinner (5-6pm), and can have other unavailabilities (e.g. appointments) already scheduled.

The Assistive Robots: We consider a set of socially assistive robots, R (r_1 … r_n), in which each robot r_l is able to execute the following activities: 1) *move* from one discrete location to another at a constant speed v_{rl}, 2) perform a *telepresence* session with a single user, 3) perform a *Bingo* session with a group of users, 4) provide a *reminder* to each user prior to a Bingo game, and 5) *recharge* its battery at a charging station. Since battery consumption depends on the activity, whenever the robot, r_l, executes an activity, its *battery level*, bl_{rl}, must remain within certain bounds (i.e., $bl\_min_{rl} <= bl_{rl} <= bl\_max_{rl}$). Each aforementioned activity has a different battery consumption rate, cr (defined as V/min): cr_move_{rl}, cr_telep_{rl}, cr_bingo_{rl}, and cr_remind_{rl}. Battery power is regained through a charging station. A constant recharging rate rr_{rl} (e.g., V/min) is used to estimate the duration of a recharging process of a robot r_l. The battery can be recharged up to bl_max_{rl}.

Charging Stations: A set of charging stations, CS (cs_1 … cs_n), exists for recharging. Each station is in a fixed location and can accommodate at most one robot at a time.

Telepresence Sessions: A set of *telepresence sessions*, S (s_1 … s_n), must be scheduled during the day. Each session s_y is characterized by: 1) the user u_k; 2) the duration, dur_{sy}, (e.g., 30 min); and 3) the time window(s) it can occur in. The session should always take place in a user's room (l_{uk}).

Bingo Games: A set of *Bingo games*, G (g_1 … g_n), must be scheduled during the day. For each game g_z, the robots will assign, find, and remind users prior to the game and, then, facilitate Bingo in a specific location, the games room (l_{game}), at the scheduled time. Only one game can occur at a time. Only one robot can conduct the game, but the robots can collaborate to deliver the reminders. Each game g_z is characterized by: 1) the duration of the game, dur_{gz}, (e.g., 60 min) and of the reminder, dur_remind_{gz} (e.g., 2 min); 2) the overall set of interested participants; 3) the minimum and maximum number of participants, p_min_{gz} and p_max_{gz}; and 4) the time window(s) in which it can occur.

The exact group of participating users of a game g_z is not known a priori nor is the time of each game, only the overall set of participants that have expressed interest in playing a game is known. Users are assigned to each game, and games are scheduled to fit the users' availabilities. Reminders must be delivered to *all* assigned users when they are available before the game starts. It is assumed that the users will go to the games room at the time specified.

Robot Activities: We describe below the conditions and constraints of the available robot activities.

Move to a Target Location: the robot must have enough battery power to reach the target location l_j from its current location l_i. The battery consumption and the duration of the activity are $(d_{i,j}/v_{rl})\times cr\_move_{rl}$ and $d_{i,j}/v_{rl}$, respectively.

Recharge battery: the robot has to be in a location with an idle charging station and the battery level has to be less than the battery capacity, $bl_{rl} < bl\_max_{rl}$. The duration of the activity is $(bl\_max_{rl}\text{-}bl_{rl})/rr_{rl}$.

Perform Telepresence Session: the robot has to be in the private room of the specified user, who must be available during the entire duration (dur_{sy}) of the activity. The battery consumption of the activity is $dur_{sy}\times cr\_telep_{rl}$.

Facilitate Bingo Game: the robot must be in the games room, with no other game underway; and all invited users must be available during the entire duration (dur_{gz}) of the game. Users must have been reminded before the game starts. The battery consumption of the Bingo activity is $dur_{gz}\times cr\_bingo_{rl}$.

Remind User: the robot has to be in the same location as the user, who cannot be interacting with another robot and must be available during the entire duration (dur_remind_{gz}) of the activity. The battery consumption of the reminder activity is $dur\_remind_{gz}\times cr\_remind_{rl}$.

For all the activities (except recharging) the robot has to have enough battery power to reach a location that has a charging station after the activity is completed.

Input and Goals: The *input* of the problem is the sets of locations *L,* users *U* (including their corresponding profiles), charging stations *CS*, available robots *R* (with their initial location and corresponding velocity, battery levels and limits, and rates), and the requested telepresence sessions *S* and Bingo games *G* with their corresponding properties. The *goal* is to have a plan of robot activities in which all the requested telepresence sessions and Bingo games are scheduled. All robots must be at a recharging location at the end of the day.

Multi-Robot and Multi-User P&S System

In this work, we address the challenges of integrating P&S of the activities of multiple robots while considering the individual schedules of multiple users and the constraints of the proposed problem. We propose a modular centralized P&S system architecture, Figure 1, for planning, scheduling, executing and monitoring the daily activities of a set of socially assistive robots assisting multiple residents in a retirement home setting. The system is composed of two main components: 1) a *centralized server* which is designed to receive users' schedules, room availability and activity requests (telepresence sessions and Bingo games), and to autonomously plan the activities of the robots throughout the day; and 2) a set of *robot on-board controllers* that are able to receive activity commands from the server and to autonomously navigate the environment, remind users, facilitate single-user telepresence sessions and multi-user Bingo games, and recharge when necessary.

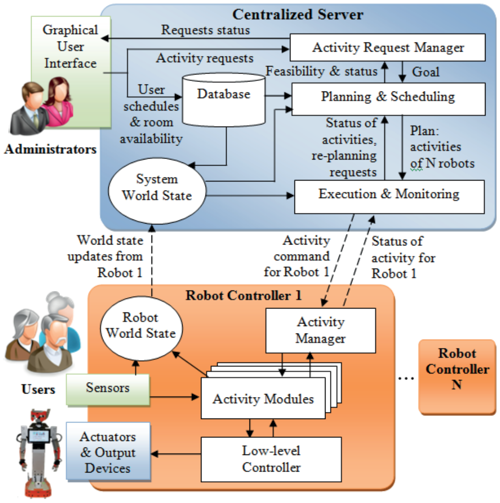

Figure 1. The Multi-Robot and Multi-User P&S System Architecture.

Centralized Server

Graphical User Interface (GUI): Designed for retirement home administrators to provide the following input through a computer: 1) the schedules of the users and the availability of the rooms, and 2) requests for telepresence sessions and Bingo games for a set of users. The former is stored in the *Database* while the latter is provided to the *Activity Request Manager* for processing. Changes to these inputs can be made; resulting in replanning.

Database: Stores the user profiles and schedules, and room availabilities. It also stores the list of robots that can be used along with the specifications of the robots' activities, commonly defined as the *domain model*. Domain models include activity preconditions (e.g., a robot can only remind a user if he/she is available), effects (e.g., battery level is decreased after a move activity) and durations (e.g., a telepresence session is 30 minutes long).

System World State: Stores a model of the current state of the world, including the topology of the retirement home (i.e., private rooms, game rooms, common areas, corridors, locations with charging stations, and distances between locations) and the state of all the elements in the system including users, robots, rooms, charging stations, and requested activities. This module uses the spatial-temporal information from user schedules to update the user states over time. For example, if a user has an appointment from 2-3pm, the *System World State* module will update the availability and location of that user in the *System World State* module for that time window. The overall world state is represented as a set of *fluents* (facts) which hold at a particular time of the day. Examples of fluents are: the location (fluent *at*) and availability (fluent *available*) of a user; the current battery level (fluent *bl*) and location (fluent *at*) of a robot; the availability of a charging station (fluent *idle*); the duration of a telepresence session (fluent *dur*) and the fact that a user has been assigned to an upcoming Bingo game (fluent *assigned(user,game)*).

Activity Request Manager: Manages the telepresence and Bingo requests sent by the administrators. A set of new activity requests is sent as a *goal* to the *Planning & Scheduling* module in order to check if a feasible plan can be obtained to fulfill the requests. Such a goal refers to the completion of the activities in the respective specified time windows and locations. If a plan is found, the status of the overall set of activity requests is: 1) monitored by the *Activity Request Manager* until the requests are completed, and 2) provided back to the administrators as feedback.

Planning & Scheduling (P&S): This module is responsible for autonomously performing both planning and scheduling of the activities for the robots to achieve the specified goal, as shown in Figure 1, based on the following input:

1) Telepresence and Bingo activity requests.
2) The user schedules and room availabilities.
3) The specifications of the robots' activities.
4) The world state.
5) The current status of the robots' activities.

In this work, the *P&S* module has been designed to incorporate off-the-shelf temporal planners (e.g., heuristicbased and timeline-based planners) that can be plugged into the module to perform the automated reasoning about activities. Depending on the planner, the inputs into the planner have to be properly parsed into the appropriate domain model and problem descriptions. The target planner is responsible for generating a *sequence of temporally annotated robot activities* (a *plan*) to solve the problem. Activities being implemented by the robots can be interrupted if needed. Namely, activities can be defined to be *interruptible* (e.g., move) or *non-interruptible* (e.g. Bingo game, telepresence session). The planner uses the *interruptibility* to determine if currently executing activities should be preempted or continued during replanning. A replanning-from-scratch approach is used when replanning is needed. The generated plan is sent to the *Execution & Monitoring* module for further execution.

Execution & Monitoring (E&M): Manages and monitors the execution of the plan. It is responsible for: 1) sending activity commands to the appropriate robots at the appropriate times, and 2) monitoring the status of these activities. A server clock is used to determine when an activity should be sent to a robot. In each cycle of the server clock, the module monitors the overall activity status set through the feedback provided by each robot's *Activity Manager*.

Before sending an activity command to a robot, the module checks if the activity's preconditions hold in the current world state to confirm the activity's applicability (e.g., checking if a charging station is available before requesting a recharging activity). In this work, low-level execution requirements (e.g., if the robot's sensors are functioning properly at an activity's start time) are handled in the robot on-board controller. If the preconditions are satisfied, the activity command is sent to the robot, otherwise, a replanning request is sent to the *P&S* module. If a robot informs the *E&M* module that an activity has failed during execution (e.g., the robot cannot reach a target location), the module informs the *P&S* module that replanning is needed. The *E&M* module can proactively check if certain conditions hold in the world state during execution of robot activities to detect potential replanning scenarios (e.g., the robot's battery level drops below safe levels). When a robot's *Activity Manager* informs the *E&M* module that an activity has finished successfully, the module verifies the expected effects in the world state based on the domain model definition (e.g., after the move activity the module checks if the robot is at the target location).

Robot On-Board Controller

Activity Manager: Manages the activity commands received from the centralized server and monitors the physical execution of each activity. The module can identify which *Activity Module* is able to perform the low-level actions to execute the activity command received from the server. Similar to the *E&M* module, the *Activity Manager* monitors the progress of the execution of a particular command. The progress of the executing activity is sent to the server as the activity status. If the server requests that an activity has to be preempted, the *Activity Manager* transfers this request to the corresponding *Activity Module*.

Activity Modules**:** Each robot has a set of *Activity Modules* that are able to implement its behavior and all low-level actions corresponding to the target activity. When the robot's *Activity Manager* requests an activity, the corresponding *Activity Module* checks the low-level execution requirements before initiating any actions. While executing the actions, the *Activity Module* sends updates to the *Robot World State* module with respect to the fluents of the robot or of other elements in the robot's immediate world. If an activity execution fails, the robot's *Activity Manager* is informed so that the status of the robot activity can be sent to the server for further decision making.

Robot World State: The *Robot World State* module is responsible for sending state updates about the robot and its environment to the *System World State* module. This provides the server with an updated overall model of the environment, robots, users, and requests in the system. In addition, the *Robot World State* receives, via the *Activity Modules*, fluent updates (e.g., after a reminder activity, the fluent *assigned(user,game)* holds for a specific Bingo game.

Low-level Controller: Implements the robot's behaviors (low-level actions). It consists of both Navigation and HRI layers. The Navigation layer autonomously moves the robot through the environment while the HRI layer generates the verbal and non-verbal communication (e.g., speech, gestures, graphical displays) for the robot.

Implementation

We have designed and implemented the proposed multi-robot and multi-user P&S architecture for validation in a real environment setting using the Robot Operating System platform (ROS 2014). Inter-communication between the server and robot controllers and intra-communication between individual modules (Figure 1) is achieved through ROS messages. The robots and server communicate with each other using a wireless network.

P&S Module

We have integrated the PDDL planner OPTIC (Benton et al. 2012) to generate a set of temporal annotated robot activities to reach the specified goals. Although the *P&S* module can potentially use any PDDL temporal planner, from our own investigation, we found OPTIC to be the only PDDL planner capable of handling a problem such as the one proposed in this paper with reasonable performance (Vaquero et al. 2014).

To translate the input information to a PDDL model, we designed a parser that generates an input ready problem specification for PDDL temporal planners (i.e., the *domain* and *problem instance)* based on data gathered from the *GUI*, *Database*, *System World State*, and the *E&M* modules. This is translated to a PDDL 2.2 (Edelkamp and Hoffman, 2004) domain specification using the same strategy used in the itSIMPLE tool (Vaquero et al. 2009). The parser translates the state of the world into a PDDL problem instance considering the user schedules and interruptible and non-interruptible activities. Users' schedules are represented using timed initial literals (TILs) (Edelkamp and Hoffman, 2004) by assigning their availabilities and known locations at specific time intervals.

For each non-interruptible activity the expected effects are translated as TILs in the problem instance. The resulting TILs are set to hold at the activity's expected end time. Herein, the *E&M* module records the fluents of interest between the start of an activity and the current execution time so that the planner can use them to evaluate the effects (e.g., an expected effect of facilitating a telepresence activity will result in the battery level decreasing during the activity). With respect to the interruptible activities (*move*), a predicate (*executing_move*) is used to represent that an activity has started. The PDDL model has an additional operator that allows the planner to stop a move activity (*stop_move*). If the planner stops a current move activity, then the *E&M* module will preempt this activity.

Socially Assistive Robotic Platform

The Tangy socially assistive robotic platform is used to perform the HRI activities in a retirement home setting. Tangy uses a combination of a synthesized voice, body language and gestures, and a touch screen tablet to interact with users. User identification during interaction is implemented using the OKAO^{TM} Vision software library (Omrom 2007) and is achieved based on facial shape models and face contour features utilizing images provided by an onboard 2D Axis M1031-W camera. The face models are compared to face models stored in an onboard database (which is in sync with the server's *Database*) to identify the individual users. When a user is detected, the *Robot World State* module updates the state of the user (e.g., his/her presence in the same region as the robot).

Tangy navigates the environment using the ROS navigation software package (Marder-Eppstein et al. 2010) which obtains 3D data from a Hokuyo URG-04LX-UG01 laser range finder mounted on a titling platform on the robot's base. While moving in the environment, the *Robot World State* module updates the location of the robot (fluent *at*). A 2D Logitech Pro C920 camera and an ASUS Xtion PRO LIVE sensor are used to monitor the Bingo game. More details on Tangy's sensors and activity-specific behaviors can be found in (Louie et al. 2014).

Experiments

Experiments were conducted to investigate the use of the multi-robot and multi-user P&S system architecture in a realistic environment setting with users. We evaluated its ability to: 1) plan and schedule daily activities of multiple robots to facilitate telepresence sessions and Bingo games while considering the requests and schedules of users; 2) adapt to unexpected scenarios during the execution of robot activities (e.g. replan); and 3) determine whether a physical robot can successfully execute the activities while meeting the necessary spatial and temporal constraints.

Implementation Scenarios

The following are inputs into the centralized server:

1) Floor topology which includes 4 private rooms, one games room, one common room, and one charging room.

2) Twenty potential users occupying this environment.

3) Three socially assistive robots: *robot A*, *robot B*, *robot C*. All robots have the following: bl_min = 0, bl = bl_max = 20, v = 12m/min, rr = 0.5, cr_move = 0.04 and cr_telep = cr_remind = cr_bingo = 0.1 (V/min).

4) The schedules of users, the room availabilities (the games room is scheduled for cleaning from 4-5pm) and the activity requests for telepresence sessions and Bingo games are provided by 6am that day. The requests include 6 telepresence requests by 6 distinct users, and 8 additional users requesting Bingo (2 games in total). A Bingo game must have 4 participants and occurs in the *games room*.

5) In addition to the meal times, each user has one or two 1-hour activities in his/her schedule, during which the robots cannot disturb him/her (e.g. *non-interruptible activities*). Other activities (e.g., walk in the common area, reading in a room) which allow robot interactions (*interruptible activities*) occur at least once for each user.

Between 6-7am the P&S architecture plans and schedules the activities of the 3 robots given the above. Four different execution scenarios are considered: 1) *no disturbance*- the planning scenario exactly matches the implementation scenario, 2) *environmental disturbance*- a robot does not reach a telepresence location in time due to a blocked corridor along its path; 3) *user disturbance*- a user is not found in his/her room by a robot during the reminder activity; and 4) *robot resource disturbance*- the battery level of a robot is too low to execute its next scheduled HRI activity.

Results

Our performance metrics for the P&S architecture are: 1) runtime for generating plans for the robots at both the beginning of the day and after each disturbance; and 2) the robot's ability to implement its planned activities based on the problem's spatial-temporal constraints. For all the scenarios, the *P&S* module generated the initial plans in less than 5 seconds, and replanning took less than 1 second.

The three robots were able to successfully execute all six of the requested telepresence sessions and the two Bingo games. Figure 2 shows examples of the robots performing the activities in the environment. Replanning for the disturbances occurred as follows. The *environmental disturbance* scenario required *robot A* to perform a telepresence session at 10:30am. However, *robot A* was not in the required location by 10:30am due to the corridor being blocked, and hence, a trigger was sent to the server to reassign *robot C* to perform the telepresence 12 minutes later, while *robot A* was sent to the games room to prepare for a Bingo game. In the *user disturbance* scenario, *robot C* did not find a specific user in his room at 9:30am to deliver a Bingo reminder. The reminder was rescheduled later for *robot B*. For the *robot resource disturbance* scenario (Figure 3), *robot A* was performing recharging at 11:00am, however, the robot was not docked properly with the charging station and at the end of this activity (11:10am) its battery was not fully recharged to complete an upcoming Bingo activity. *Robot A* was commanded to continue charging for an additional 10 minutes, while *robot C* was assigned to the Bingo activity. The overall results show the system is responsive to disturbances and can generate plans (replans) in real time for the robots to execute.


(a) Navigating environment (b) Assisting with telepresence (c) Bingo game reminder (d) Facilitating a Bingo game

Figure 2. Example of Robot Activities in the Environment.

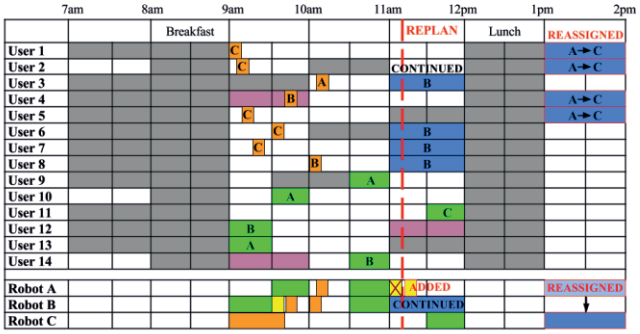

Figure 3. Robot Resource Disturbance Scenario. *A*, *B*, and *C* refer to the robots; *Gray*: user non-interruptible activities; *Purple*: user interruptible activities; *Green*: telepresence activity; *Orange*: reminder activity, *Blue*: Bingo activity; and *Yellow*: recharge activity.

Conclusion

In this paper, we have introduced and implemented a multi-robot and multi-user P&S system architecture to address the challenge of a team of socially assistive robots planning and scheduling HRI activities for multiple residents in a retirement home setting. The robots plan and schedule their activities based on the requests and schedules of residents, and the layout of the environment including room availability. Experiments with three robots demonstrate that the P&S architecture was able to plan/replan robot activities that meet the temporal and spatial constraints of the proposed problem in an environment with twenty users.

Acknowledgements

The authors thank the residents, staff and family members at the Residential Care Homes. We also thank NSERC (Natural Sciences & Engineering Research Council of Canada), TVN (Technology Evaluation in the Elderly Network, supported by the Government of Canada through the Networks of Centres of Excellence) and the Canada Research Chairs Program.

References

Bennewitz, M.; Faber, F.; Joho, D.; Schreiber, M; and Behnke, S. 2005. Towards a humanoid museum guide robot that interacts with multiple persons. IEEE Int. Conf. Humanoid Robots, pp. 418-423.

Benton, J.; Coles, A. J.; and Coles, A. I. 2012. Temporal Planning with Preferences and Time-Dependent Continuous Costs. Int. Conf. on Automated Planning and Scheduling, pp. 2-10.

Cesta, A.; Cortellessa, G.; Rasconi, R.; Pecora, F.; Scopelliti, M.; Tiberio, L. 2011. Monitoring Elderly People with the RoboCare Domestic Environment: Interaction Synthesis and User Evaluation. *Computational Intelligence*, 27(1): 60–82.

Coltin, B.; Veloso, M.; and Ventura, R. 2011. Dynamic User Task Scheduling for Mobile Robots. AAAI Workshop on Automated Action Planning for Autonomous Mobile Robots, pp. 1-6.

Edelkamp, S., and Hoffmann, J. 2004. PDDL2.2: The language for the classical part of the 4th International Planning Competition. Tech. rep. 195, Albert-Ludwigs-Universität Freiburg, Institut für Informatik.

Louie, W.G.; Han, R.; and Nejat G. 2013. A Socially Assistive Robot to Promote Stimulating Recreational Activities at Long-term Care Facilities. *Journal of Medical Devices-Trans. of the ASME*, 7: 030944-1.

Louie, W.G.; Vaquero, T.; Nejat, G.; and Beck, J.C. 2014. An Autonomous Assistive Robot for Planning, Scheduling and Facilitating Multi-User Activities. IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 5292-5298.

McColl, D.; Louie, W.G.; and Nejat, G. 2013. Brian 2.1: A Socially Assistive Robot for the Elderly and Cognitively Impaired. *IEEE Robotics & Automation Magazine,* 20(1): 74-83.

Montemerlo, M.; Prieau, J.; Thrun, S.; and Varma, V. 2002. Experiences with a mobile robotics guide for the elderly. AAAI National Conference on Artificial Intelligence, pp. 587–592.

Petrick, R.P.A., and Foster, M.E. 2013. Planning for Social Interaction in a Robot Bartender Domain. Int. Conf. Automated Planning and Scheduling, pp. 1-9.

Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; Thrun, S. 2003. Towards Robotic Assistants in Nursing Homes: Challenges and Results. *Robotics and Autonomous Syst.*, 42(3-4): 271-281.

Robot Operating System. 2014. Available: http://www.ros.org.

Vaquero, T. S.; Silva, J. R.; Ferreira, M.; Tonidandel, F.; and Beck, J. C. 2009. From requirements and analysis to PDDL in itSIMPLE3.0. Int. Competition on Knowledge Engineering for Planning and Scheduling at ICAPS, pp. 1-8.

Vaquero, T.; Beck, J.C.; and Nejat, G. 2014. Planning and Scheduling Single and Multi-Person Activities in Retirement Home Settings for a Group of Robots. Int. Conf. on Automated Planning & Scheduling Workshop on Planning and Robotics, pp. 59-68.

Smith, D.E.; Frank, J.; and Jonsson, A.K. 2000. Bridging the Gap Between Planning and Scheduling. *Knowledge Engineering Review*, 15(1): 1-34.