Integrating Socially Assistive Robot (SAR) and Smart Home to Support Caregiving of Individuals with Dementia Disease

2019 Summer Report
Tianyi Gu
Content

The Problem

Smart Home Design

The Robot

The AI Planner

Results
The Problem

How to Taking good care of People with Dementia:

- Health
- Well-being
- ...

Family member experience Care Burden

Or

$50,000 / year for a home health aide

$100,000 / year for a 1bd nursing home
The Problem

How to Taking good care of People with Dementia:

- Health
- Well-being
- ...

Family member experience Care Burden

Or

$50,000 / year for a home health aide

$100,000 / year for a 1bd nursing home

Can robot and current available technologies help?
Smart Home Design
The Robot
The Robot

Ros Nodes:

- Mapping and Localization: gmapping, amcl
- Auto Navigation: move_base
- Face Module: face-detector, face_recognition
- Task Planning: ROSPlan
- Action Service Nodes
- Executive
The AI Planner

ROSPlan:

- PDDL
- Interfaces available for many planners

We use:

- PDDL 2.1
- Contigent-FF
The AI Planner

SHR Domain:

```prolog
(define (domain shr_contingent)
  (:requirements :strips :typing :disjunctive-preconditions)
  (:types
   landmark
   robot
   message
   sensor
  )
  (:predicates
   (robot_at ?v ?l - landmark)
   (is_home ?l - landmark)
   (notified ?msg - message)
   (message_at ?msg - message ?l - landmark)
   (is_on ?ss - sensor)
   (is_off ?ss - sensor)
   (available_to_check_s ?ss - sensor)
   (sensor_after_notified ?ss -sensor ?msg - message)
   (is_safe)
   (is_not_safe)
  )
  (:actions
   ; Move to any landmark, avoiding terrain
   (action move_to landmark
     :parameters (?v - robot ?from ?to - landmark)
     :precondition (robot_at ?v ?from)
     :effect (and
       (robot_at ?v ?to)
       (not (robot_at ?v ?from)))
   )
   ; Notify message at landmark
   (action notify_at
     :parameters (?v - robot ?l - landmark ?msg - message)
     :precondition (and
       (robot_at ?v ?l)
       (message_at ?msg ?l))
     :effect (and
       (forall (?ss - sensor) (when (sensor_after_notified ?ss ?msg) (available_to_check_s ?ss)))
       (notified ?msg))
   )
   ; Check if sensor ss is on
   (action check_sensor_on
     :parameters (?ss - sensor)
     :precondition (available_to_check_s ?ss)
     :observe (is_on ?ss)
   )
  )
)
```

Mid-night Problem:

```prolog
(define (problem task_conditional)
  (:domain shr_contingent)
  (:objects
   door_home - landmark
   pioneer - robot
   midnight_warning - message
   leaving_home - message
   doors - sensor
  )
  (:init
   (robot_at pioneer home)
   (is_home home)
   (message_at midnight_warning doors)
   (sensor_after_notified doors midnight_warning)
   (known (is_on doors))
   (known (is_off doors))
   (is_safe)
   (is_on doors)
   (is_off doors)
  )
  (:goal (is_safe))
)
```
Mid-night Task Plan:

Start of a Planning Cycle

1. Monitoring DB (M2)
   - Move to the door
   - Notify Mid-night Msg
   - Monitoring DB (door)
   - Yell Patient Name

2. Play Caregiver’s Video
   - Call Caregiver (notification)
   - Monitoring DB (M1)
   - Call Caregiver (alarm off)

3. Call Caregiver (alarm off)

4. Call 911

Finalise success

End of a Planning Cycle

Finalise fail

Call 911
Result

Tested with Actor Patient and Caregiver:

- Midnight scenario with 4 situation - all success
- Medication scenario with 4 situation - all success

Sep 21:

- Invite Real Caregivers to try it
Future work

Online planning deal with uncertainty:

- Adaptive replanning with nested contingent planner

User-Friendly Design:

- domain knowledge engineering for PDDL