



Interreg España - Portugal



Fondo Europeo de Desarrollo Regional
Fundo Europeu de Desenvolvimento Regional



Iniciativas innovadoras para el impulso del envejecimiento activo en la región EuroACE

TÍTULO DOCUMENTO

Estudio de la interacción humano-robot

RESPONSABLE:

RoboLab

FECHA DE ENTREGA:

31-04-2019

AUTORES:

Pedro M. Núñez Trujillo

CONTRIBUCIONES:

José Luis Moyano, Blas Pagador, Pedro Núñez, Paulo Gonçalves, Rui Rocha, Carolina Vila

RESUMEN DEL DOCUMENTO

Este documento detalla los métodos y sistemas de interacción humano-robot. Definición de las características del movimiento humano que permiten inferir la actividad desarrollada por el paciente durante las sesiones asistenciales. Descripción de los modelos de conocimiento durante una HRI.

Información del documento

FECHA	VERSIÓN	COMENTARIOS
15/09/2018	V0.0	Versión inicial
22/02/2019	V1.0	Versión con contribuciones de Pedro Núñez, Paulo Gonçalves, Rui Rocha
31/02/2019	V2.0	Versión final



Fundación Centro de Cirugía de
Mínima Invasión Jesús Usón
(CCMIJU, España)

Coordinador:
Jose Blas Pagador
jbpagador@ccmijesususon.com



Instituto Politécnico
de Castelo Branco

Instituto Politécnico de Portalegre.
(IPCB, Portugal)

Contacto:
Paulo Gonçalves
paulo.goncalves@ipcb.pt



Universidad de Coimbra (UC,
Portugal)

Contacto:
Rui Rocha
rprocha@deec.uc.pt



Cluster de la Salud

Cluster Sociosanitario de
Extremadura (Cluster, España)

Contacto:
Nicolas Montero
gerente@clustersalud.es



Universidad de Extremadura (UEX,
España)

Contacto:
Pedro Núñez
pnuntru@unex.es



Instituto Politécnico de Guarda (IPG,
Portugal)

Contacto:
Carolina Vila-Cha
cvilacha@ipg.pt

ÍNDICE

RESUMEN EJECUTIVO

1. Sistema de interacción del robot

[1.1. Aproximaciones de diseño](#)

[1.2. Aspectos específicos del diseño](#)

[1.3. Embodiment](#)

2. Sistema de percepción de humanos en el entorno

3. Modelos de conocimiento durante una interacción humano-robot en EuroAge

RESUMEN EJECUTIVO

Este documento detalla los métodos y sistemas de interacción humano-robot. Definición de las características del movimiento humano que permiten inferir la actividad desarrollada por el paciente durante las sesiones asistenciales. Descripción de los modelos de conocimiento durante una HRI en un caso de uso específico.

1. Sistema de interacción del robot

El campo de la Interacción Hombre-Robot (Human-Robot Interaction, HRI) se dedica a estudiar, diseñar y evaluar sistemas robóticos que conviven o son usados por personas. El término en sí mismo implica la existencia de una comunicación entre el robot y la persona, que podrá adoptar distintas formas en función de la distancia entre ambos. Por ello, la interacción se ha separado en dos categorías generales:

- Interacción remota, en la que la persona y el robot están distantes, no sólo físicamente, sino que también podría ser una distancia temporal (por ejemplo, se puede interactuar con el Mars Rovers, pero la distancia no será sólo física, sino también temporal).
- Interacción próxima, en la que persona y robot comparten un mismo espacio y secuencia temporal.

Dentro de estas categorías resulta práctico distinguir entre aplicaciones que requieren movilidad, manipulación física o interacción social. Específicamente, la interacción social incluye los aspectos social, emotivo y cognitivo de la interacción. Así, en una interacción social, el robot y la persona se interrelacionan como iguales o compañeros, lo que la hace más cercana a la interacción próxima definida anteriormente¹. La interacción social requiere que el robot posea capacidades que le permitan comunicarse con la persona a través de canales verbales (diálogo) y no verbales (expresiones faciales, gestos y posturas), pero también que le permitan aprender nuevas habilidades y ejecutar tareas cooperativas (aunque no exista siempre el contacto físico). Surge además la necesidad de integrar el modelado de las tareas y primitivas motoras con la interpretación de la intención de la persona, para así poder reconocer actividades vitales existentes o novedosas. Un robot social deberá finalmente estar preparado para relacionarse no sólo con usuarios expertos, sino también con personas que nunca antes hayan tratado con un robot como él. Para conseguir que su presencia sea aceptada, el robot deberá poder desplegar un amplio abanico de capacidades, entre las que serán importantes su aspecto y respeto por las mínimas reglas de comportamiento social.

A continuación se analizan los aspectos más importantes en la metodología a seguir para el diseño de un robot socialmente interactivo. Aunque se presenten aisladamente, muchos de ellos están íntimamente interrelacionados.

¹ El concepto de interacción está actualmente empezando a restringirse a un proceso en el que se intercambian mensajes o comandos, lo que en este documento definimos por interacción social comienza a ser denominado **colaboración**.

1.1. Aproximaciones de diseño

Las personas somos, normalmente, expertas en la interacción social. Por ello, si la tecnología es capaz de responder correctamente a las expectativas sociales de la persona, ésta encontrará que la interacción es fácil e intuitiva. En esta búsqueda por hacer la interfaz lo más parecida posible a la que normalmente empleamos, muchos investigadores han explorado el espacio de diseño ofrecido por la apariencia humana (o de algunos animales), intentando así que el robot ofrezca, y haga surgir de forma espontánea en la persona, comportamientos propios de aquellos a los que trata de imitar. Es por ello que los robots se dotan de caras, capacidades para dialogar o reconocer expresiones. El objetivo parece siempre ser que la interacción hombre-robot sea lo más cercana posible a una relación entre personas (o, al menos, a la relación que la persona podría tener con una mascota).

Desde la perspectiva del diseño, los robots socialmente interactivos se pueden clasificar como inspirados en dos estrategias diferentes. Por un lado, el diseño biológicamente inspirado persigue que el robot imite, o simule internamente, el comportamiento social que aparece en determinadas especies animales. Es una aproximación en la que se tratan de imitar las capacidades cognitivas de los animales, por lo que las bases en las que se sustenta el diseño se encuadran en disciplinas como la psicología, la neurociencia o la biología. El diseño orientado a la funcionalidad pretende que el robot exhiba como fin un comportamiento socialmente correcto, pero su esfuerzo es conseguir ese resultado externo. Esto es, las bases, que se fundamentan en procesamiento de señal, planificación clásica, etc..., no tienen porque sustentarse sobre una arquitectura en la que la se entienda o justifique internamente cómo se genera este comportamiento socialmente correcto.

1.2. Aspectos específicos del diseño

El diseño de cualquier robot autónomo móvil, sea socialmente interactivo o no, debe resolver un determinado número de problemas comunes, sean a nivel cognoscitivo (planificación, toma de decisiones), perceptivo (navegación, sentido), de actuación (movilidad, manipulación), interactivo (interfaces de usuario, dispositivos de entrada o de salida) y de arquitectura (control, sistema electromecánico). Además de todos los enumerados, un robot socialmente interactivo deberá resolver problemas adicionales, relativos a la interacción social. Entre estos problemas específicos destacan:

- Percepción orientada a la persona. Un robot socialmente interactivo deberá percibir e interpretar, mayoritariamente, la actividad y comportamiento de personas. Esto incluye detectar y reconocer gestos, monitorizar y clasificar actividades o comportamientos, predecir intenciones o roles sociales...
- Interacción hombre-robot natural. Para conseguir que personas y robots se comuniquen como iguales, el robot deberá actuar a través de canales de comunicación naturales a la persona, como pueden ser el diálogo, la expresión facial o la gesticulación, así como seguir las reglas socialmente impuestas.
- Canales sociales entendibles. Un robot socialmente interactivo deberá proporcionar a la persona señales que le permitan a éste tanto conocer cual es el estado interno del robot como interactuar de una forma ágil e intuitiva. Como se ha comentado, estos canales incluyen factores verbales y no verbales, como la expresión facial, las posturas y los gestos.

- Respuesta en tiempo real. Los robots socialmente interactivos deben responder en la interacción a una velocidad similar a las personas. Así, un robot social necesitará llevar a cabo, de forma simultánea, distintas tareas, encaminadas no solo a guiar su comportamiento, determinando cual es la siguiente acción a realizar, sino también para mostrar atención e intencionalidad, así como para manejar correctamente la interacción social con las personas en su entorno.

1.3. Embodiment

Difícilmente traducible, la personificación (embodiment) es un concepto que aparece en robótica en la última década, y que hace referencia a todo aquello que permite establecer una base con la que acoplar un sistema con su entorno, creando para ello potenciales de actuación (perturbación) mutua. En el Diccionario de la Real Academia Española de la Lengua, el término personificar implica la acción de ‘atribuir vida o acciones o cualidades propias del ser racional al irracional, o a las cosas inanimadas, incorpóreas o abstractas’. En cierta forma, ambos conceptos se complementan en nuestro marco de trabajo. El embodiment de un robot será mayor, cuanto mayor sea su capacidad para actuar (perturbar), o ser perturbado, por el entorno. Esto es, será mayor cuanto mayor sea su capacidad para llevar a cabo acciones propias de un ser racional. Es esta medida de la interacción con el entorno la que evalúa normalmente la actual evolución de la robótica. En un marco de interacción social, si la capacidad de personificarse de un robot es elevada, si sus canales para influir/ser influido sobre/por el entorno son variados y eficientes, el robot estará más cerca de ser aceptado por las personas con las que, en un futuro cercano, deberá compartir un mismo entorno. El aspecto queda de relieve en la Fig.1, en la que se muestra un robot con una capacidad de personificación ya importante, pero limitada (el Nomad200 de ActiveMedia), y otro (el iCub) con gran cantidad de canales para perturbar o ser perturbado por el entorno.

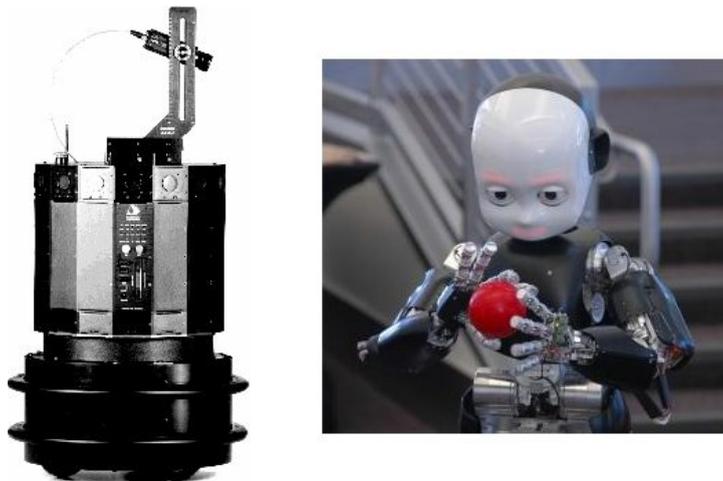


Fig. 1. El Nomad200 (a la izquierda) es un robot con un *embodiment* importante (sensores s3nar e infrarrojos, c3maras, *bumpers*...) pero limitado. El iCub (derecha) es un robot con una personificaci3n muy acusada

El valle inexplicable de Mori

Según el principio de la robótica descrito por M. Mori, el valle inexplicable es una reacción emocional de rechazo, por parte de los humanos, contra aquellas entidades no humanas que guardan un parecido tanto físico como de conducta al de la persona.

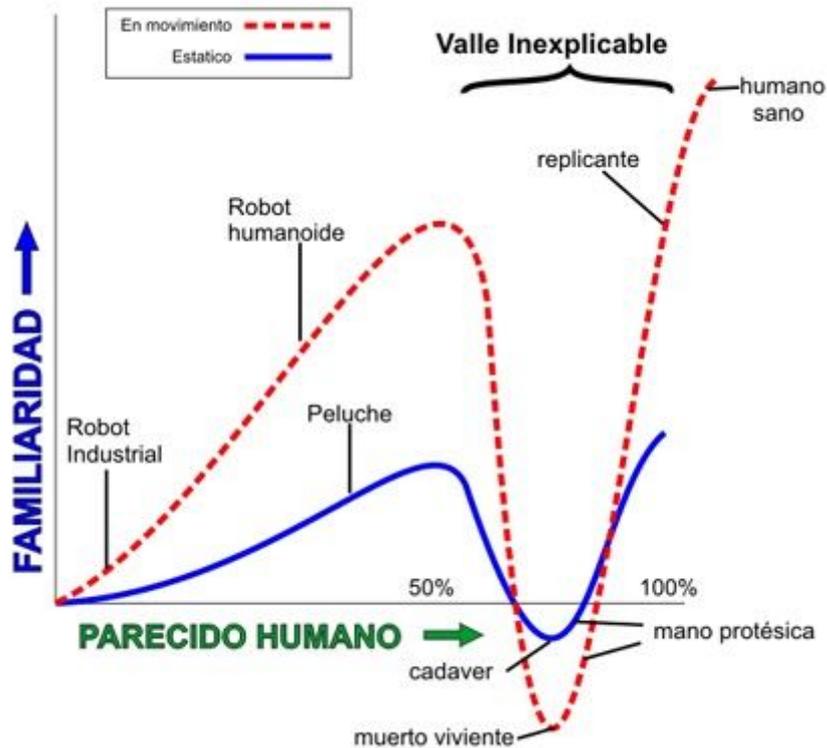


Fig. 2. El valle inquietante o inexplicable de Mori

(<http://www.santiagokoval.com/2009/05/23/el-valle-inquietante-de-masahiro-mori/>)

Una explicación a este fenómeno puede ser que, en el caso que la entidad se parezca bastante a un humano, sus características "no humanas" resaltan más, creando así una sensación de lejanía o extrañeza, y provocando un sentimiento de rechazo. Otra indica que este comportamiento se debe a que algunas acciones y ciertas características de los robots, se asemejan a las de los enfermos y moribundos, pero que al no tener causa concreta del motivo de este comportamiento, pueda crear en nuestra mente la sensación de riesgo contra nuestra propia integridad, así como poner en una paradoja nuestra lógica inconsciente. Un muy buen ejemplo de este fenómeno fuera del campo de la robótica es el de la bestia de Frankenstein. Su comportamiento, rasgos y origen eran claramente humanos pero hacían destacar a los "no humanos". Las cicatrices y su origen necrótico alejan a la criatura de la emoción humana de la afinidad.

2. Sistema de percepción de humanos en el entorno

A lo largo del proyecto EuroAGE los socios de la Actividad 2 han trabajado de forma conjunta en el despliegue de las habilidades de sus robots sociales. Entre estas habilidades y cómo afectan al comportamiento social del robot, se encuentra la detección de las personas en el mundo. Una correcta detección, así como un mejor seguimiento o tracking representan el éxito de muchas de las acciones que el robot lleva a cabo y su aceptación por parte de las personas que van a trabajar con él (no solo personas mayores, también cuidadores, familiares, etc).

A continuación se describe el método seguido para detectar humanos en el entorno:

2.1 Detección de humanos en el entorno

Los sensores empleados para la detección de humanos han sido cámaras Orbbec Astra Pro. Se tratan de cámaras 3D con un rango de visión de 0.6 m a 8m (Rango óptimo de 0.6 - 5m). Dichos sensores ofrecen imágenes de profundidad y RGB con gran resolución. Algunas de las ventajas de Orbbec es que ofrecen un gran número de herramientas en su SDK (Software Development Kit), como por ejemplo, su software de detección y tracking de personas. Actualmente, dicho software es capaz de detectar hasta 5 personas en el entorno y ofrecer una lista de joints para cada una (puntos que componen su esqueleto).

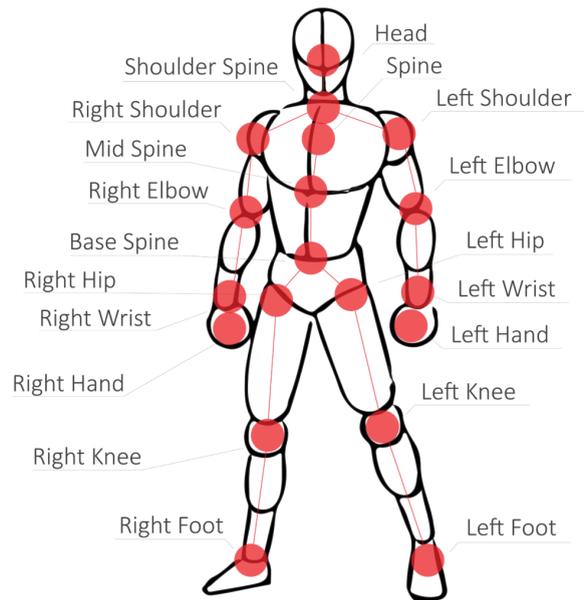


Fig. 3. Esqueleto definido según el modelo Orbbec de ASTRA

2.2 Multi-Modal Tracking

Para mejorar la precisión y la solidez del algoritmo de detección y tracking, se ha utilizado una mezcla de diferentes tipos de tracking. En este trabajo se utilizan dos algoritmos diferentes: uno de ellos es el tracking corporal tridimensional proporcionado por Astra. En la Fig. 3 se pueden ver gráficamente las articulaciones obtenidas por este algoritmo. El segundo es un típico algoritmo de detección y seguimiento de rostros en 2D. En la Fig. 4 puede verse gráficamente la detección de una cara usando este algoritmo. La información obtenida con ambos se fusiona para estimar la postura de cada humano en el entorno.

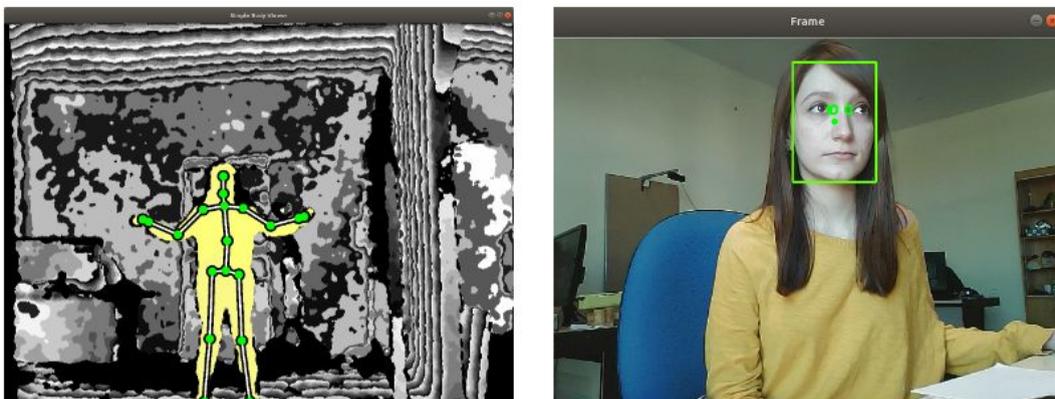


Fig. 4. Esqueleto detectado y cara detectada por el algoritmo

El tracking facial utilizado se implementa con OpenCV (Open Source Computer Vision Library).

La detección de rostros se realiza con el detector de rostros basado en algoritmos de aprendizaje profundo, disponible en OpenCV versión 3.3 o superior. Como resultado de la detección, se obtienen los cuadros delimitadores de todas las caras detectadas para cada fotograma de vídeo. Se calcula el centro de cada cuadro delimitador y se le asigna un id. Si en el siguiente fotograma de vídeo aparece un nuevo cuadro delimitador, se calcula su centro y se comprueba si corresponde a alguna de las personas presentes, calculando la distancia euclidiana con los centros obtenidos en el fotograma anterior. Si coincide, el id será el del fotograma anterior, en caso contrario se le asignará uno nuevo.

Dado que cada algoritmo de tracking empleado asigna un identificador a cada persona detectada, es necesario cotejar las personas detectadas en cada uno de ellos para obtener un identificador genérico. Sea el $B_id=\{idb1,idb2,\dots,idbn\}$ a lista de identificadores asociados a diferentes personas detectadas (y a las que se les realiza el tracking) por el algoritmo de seguimiento corporal. Además, sea $face_id=\{idf1,idf2,\dots,idfm\}$ la lista de identificadores asociados a los diferentes humanos detectados (y a los que se les realiza el tracking) por el algoritmo de rastreo facial. A continuación, se necesita una etapa de correlación entre ambas listas de humanos.

Para realizar el cotejo entre la información obtenida por los dos algoritmos de tracking, se debe utilizar el conjunto de todas las cajas delimitadoras obtenidas en la detección facial. Para cada $id_bi \in B_id$, y para cada $id_fj \in face_id$ se comprueba si las coordenadas de los píxeles de la articulación correspondiente a la cabeza, JH, está contenida dentro de los límites del cuadro delimitador $B[idfj]$. Si esto ocurre, se considera que es la misma persona que la $idfj$

Entonces, el algoritmo estima las poses humanas como se explica a continuación:

Cálculo de la posición

Para estimar la posición de la persona, primero se obtiene la lista de articulaciones de su tronco, $J_{trunk}=\{JH,JN,JSS,JMS,JBS\}$, siendo H,N,SS,MS yBS la cabeza, el cuello, el hombro, el cuello, las caderas, respectivamente.

Cada $J_i=[x,y,z]$ son las coordenadas 3D de la articulación en el marco de referencia de la cámara. Para obtener la posición real de la persona, es necesario transformar la posición de cada articulación en coordenadas del mundo real $j=[x,y,z]_{mundo}$. Para ello se utiliza InnerModelclass de RoboComp, ya que proporciona las herramientas necesarias para transformar puntos en diferentes marcos de referencia. Por cada hi humano detectado en el entorno, su posición $hi[x,z]_{mundo}$ se calcula como la media de las posiciones x y z de las articulaciones del tronco. Puede ocurrir que alguna articulación no se encuentre pero que al menos una de las articulaciones se encuentre, de lo contrario no se puede realizar la estimación de la postura humana. El número de articulaciones encontradas para cada persona puede interpretarse como una medida de la confianza de los datos obtenidos para esa persona.

Cálculo de la rotación

La rotación de la persona se estima en base a la posición de sus hombros. Es necesario obtener las articulaciones de ambos hombros, JLS y JRS, (es decir, el hombro izquierdo y el hombro derecho), para realizar el cálculo. De manera similar a la estimación de la posición, la información obtenida se encuentra en el marco de referencia de la cámara $J_i=[x,y,z]_{Astra}$. El modelo interno también se utiliza para transformar las coordenadas de las articulaciones en el

marco de referencia del mundo $J_i = [x, y, z]_{\text{mundo}}$

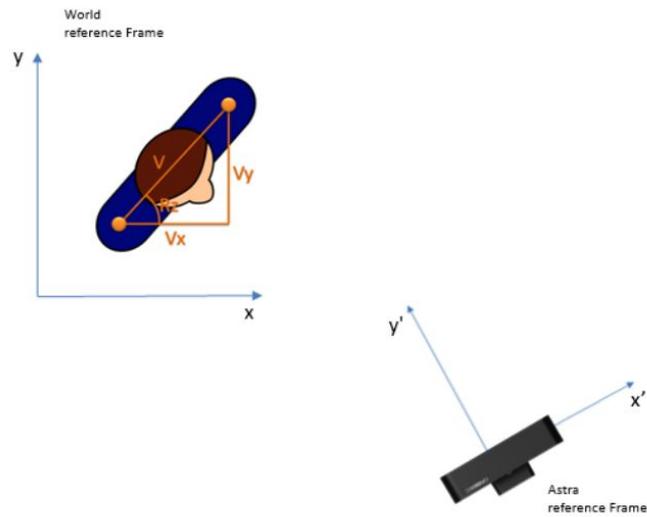


Fig. 5. Parámetros utilizados en el cálculo de la rotación. También se ilustran los diferentes marcos de referencia de Astra y World

El vector V que conecta ambos hombros se calcula como se indica en el ec. 5.1. Dado que en el mundo la coordenada y representa la altura, para el cálculo de V sólo se utilizan x y z , siendo $V_i = [x, z]_{\text{mundo}}$. La rotación del eje Z es el arco tangente de la división de la posición Y del vector V por su posición X , tal como se indica en el Ec.5.2. En la Fig. 5 se representan gráficamente los parámetros utilizados para el cálculo de la rotación y los diferentes marcos de referencia.

$$V = V_{LS} - V_{RS} \quad (5.1)$$

$$R_z = \arctan \frac{V_y}{V_x} \quad (5.2)$$

Después de calcular la posición y la rotación de una persona, esta información se incluye en el modelo interno del robot (i.e., DSR), para que su información pueda ser utilizada por otros agentes de la arquitectura.

Una vez que se ha logrado una detección y seguimiento fiables, el objetivo es mejorar el algoritmo de navegación actual, convirtiéndolo en un algoritmo en el que el espacio personal de los humanos puede adaptarse al entorno. Las mejoras propuestas para lograr este objetivo se explican en las secciones siguientes.

2.3 Tracking multicámara

El sistema descrito anteriormente hace uso únicamente de la cámara del robot. En los entornos AAL, con un robot SAR integrado en el ALab, el uso de multicámara puede ofrecer unos resultados más robustos. En este sentido, a lo largo del proyecto EuroAGE se han testeado diferentes soluciones aportadas por los diferentes grupos de investigación participantes.

Un tracking multicámara requiere cierta superposición de imágenes. El tamaño de esta ventana de superposición está directamente relacionado con el tiempo de cómputo, por tanto su selección debe realizarse con un estudio previo que minimice este tiempo sin perder robustez en la detección y seguimiento de personas. A lo largo del proyecto se han utilizado librerías conocidas como Yolo 9000, OpenPifPaf u OpenPose. Todos ellos son algoritmos implementados sobre redes neuronales que obtienen un esqueleto y su posición en píxeles sobre la imagen. La Fig. 6 muestra el esqueleto extraído a partir de la imagen del ALab del grupo RoboLab. La nueva versión de este agente usa cámaras RGB y la librería open PifPaf para extraer estimaciones de la posición de un humano 2D en la imagen. Se está comenzando el uso de redes neuronales de grafos para inferir la posición de la persona. Los primeros resultados ofrecen menor error en la estimación. A partir de estas estimaciones se está elaborando un agente capaz de extraer la posición 3D de la persona en el entorno. Este mismo agente se encarga de detectar si las personas están o no interactuando entre ellos. Aquí se está avanzando también con redes neuronales de grafos. La fig. 7 muestra capturas reales del laboratorio y los esqueletos extraídos por la librería.



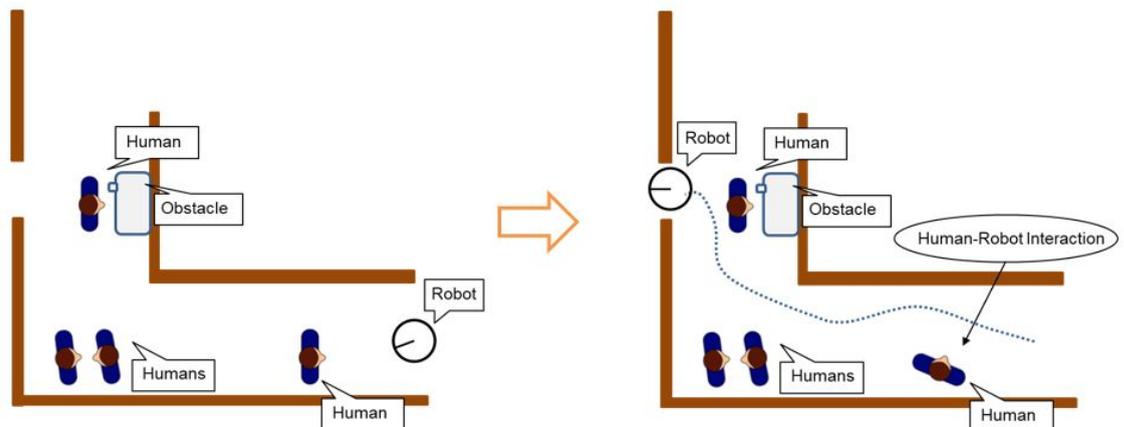
Figura 6. A la izquierda: captura de un humano en el laboratorio. A la derecha, la información del esqueleto de la persona

3. Modelos de conocimiento durante una interacción humano-robot en EuroAge

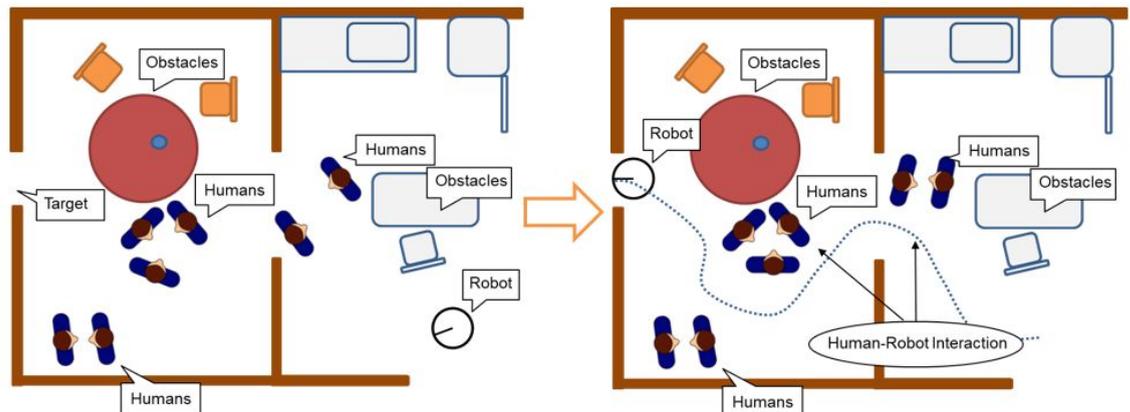
Descripción del problema

El robot es capaz de navegar en entornos con personas y grupos de personas de manera socialmente aceptable, entablando una conversación para pedir paso en tres posibles situaciones:

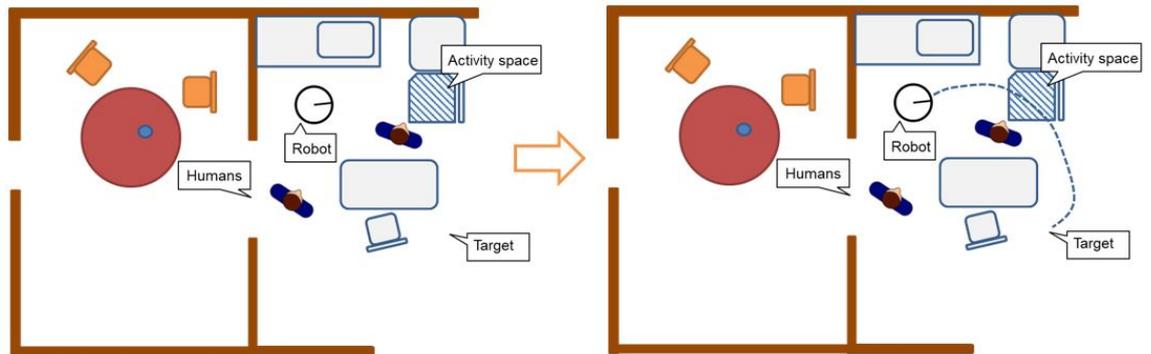
- Una persona bloquea completamente el paso (caso de uso 1)



- Un grupo de personas interactuando bloquean el paso (caso de uso 2)

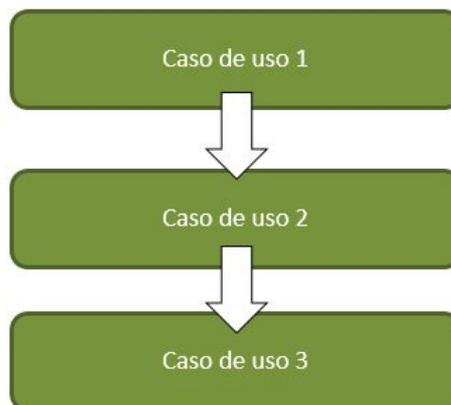


- Una persona interactuando con un objeto bloquea el paso (caso de uso 3)



El caso de uso parte de una acción “go to” a una posición determinada y termina cuando se alcanza esa posición.

Resolveremos el problema de forma secuencial, aumentando la complejidad de los escenarios



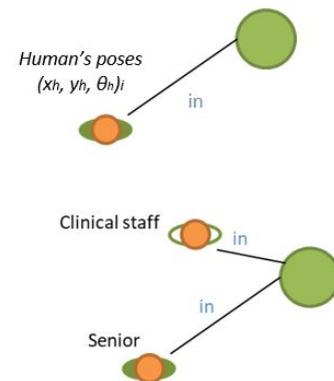
Agentes necesarios para el caso de uso 1

human agent

Agente encargado de la detección de personas en el entorno

In: cámaras

Out: lista de personas
enlaces in



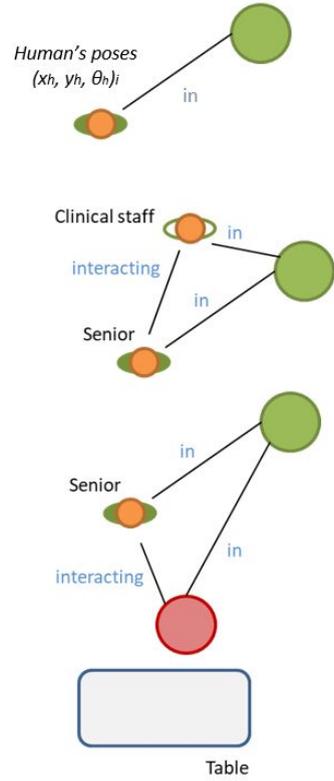
NOTA: Actualmente la posición de las personas se insertan directamente en el grafo. Para la implementación real de este agente se usará la teoría descrita en el artículo pasado del ro-man2020 (autores: Luis J. Manso, Pilar Bachiller, ...)

humanObserver agent

Agente encargado de detectar si está interactuando con otra persona u objeto del entorno: enlace is_interacting

In: lista de personas

Out:
enlaces is_interacting

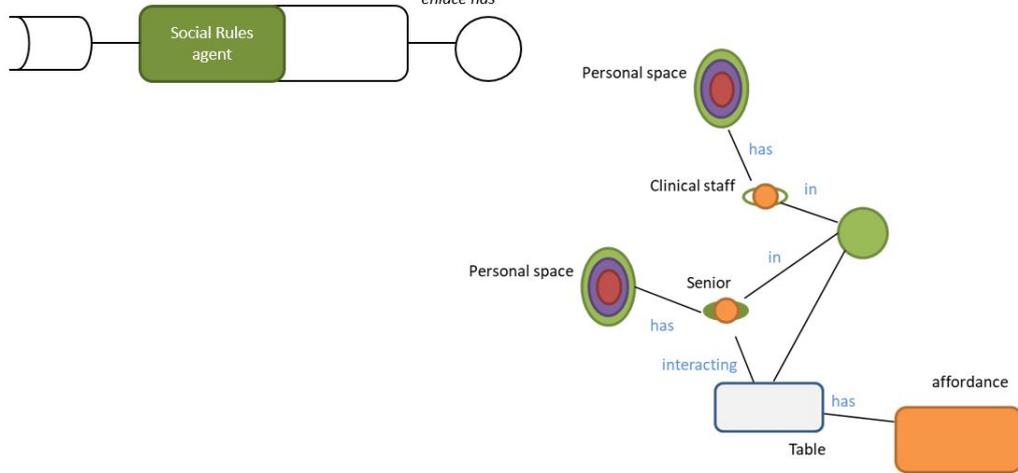


socialRules agent

Agente encargado de generar los espacios personales (íntimo, personal y social) de cada persona, encargado además de calcular la affordance del objeto (incluida también la dependencia temporal)

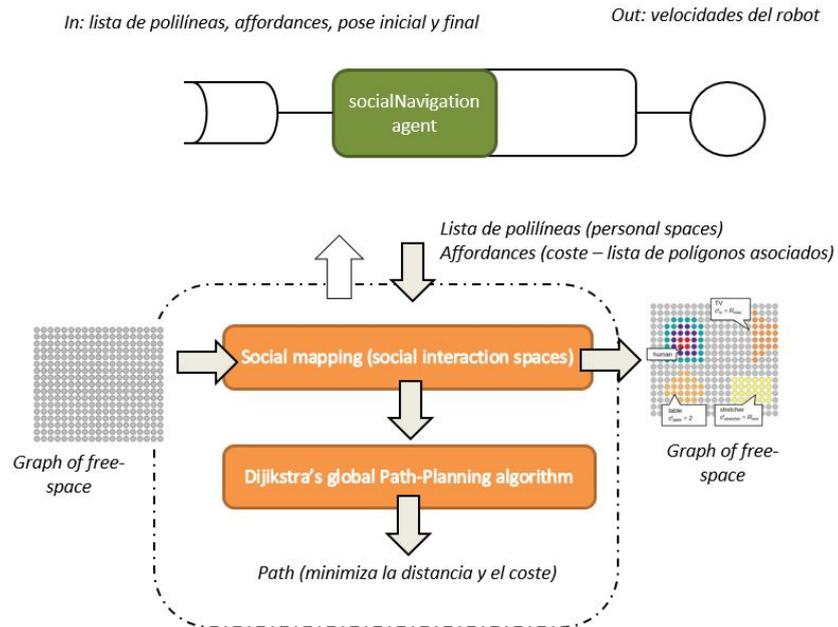
In: lista de personas, lista de objetos

*Out: lista de polígonos (personal space y affordances)
enlace has*

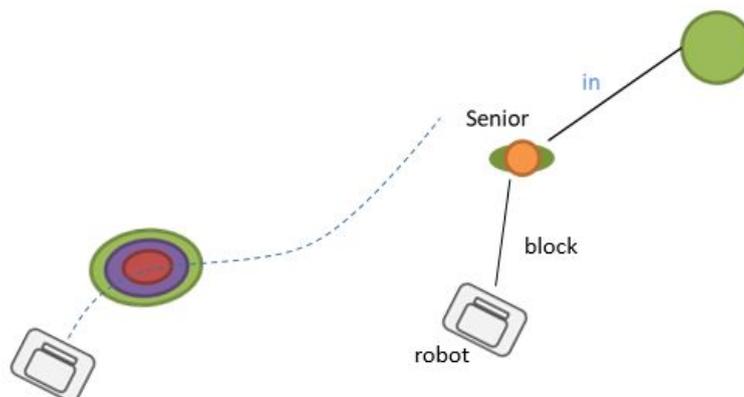


socialNavigation agent

Agente encargado de planificar la ruta social más idónea según el escenario. Tiene en cuenta el grafo de espacio libre inicial y las modificaciones que realiza en función de las polilíneas insertadas por SocialRules



NOTA: Este agente debe ser modificado para detectar una situación de bloqueo y generar el enlace correspondiente. Actualmente el grid se resetea y se van insertando las personas hasta que se bloquee el camino.



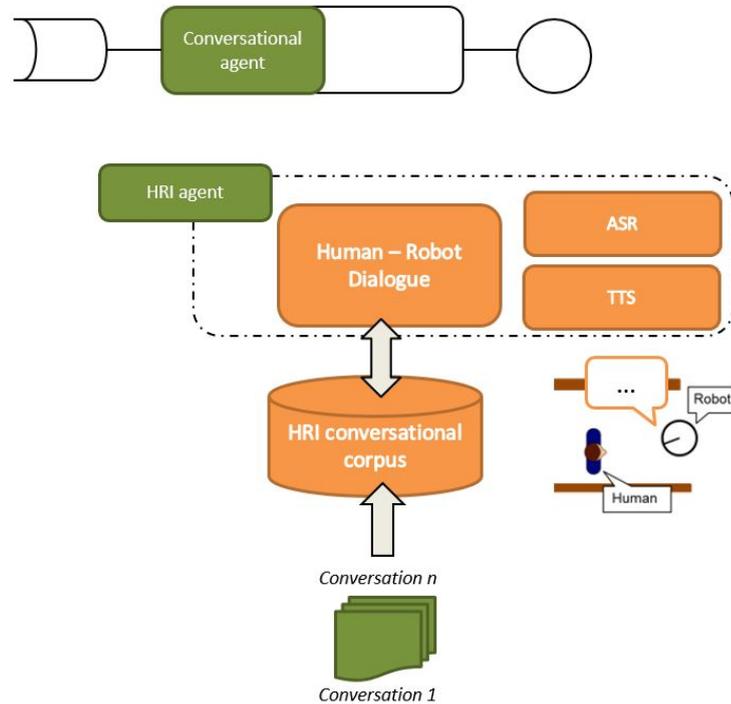
En caso de bloqueo, el robot debe re-planificar una ruta previa hacia la persona, guardando la información de la ruta final que tiene que alcanzar.

conversational agent

Agente encargado de interactuar con las personas en caso de detección de bloqueo. Debe ser lanzado una vez el robot esté cerca de la persona que bloquea, y seguir la evolución del grafo durante la conversación. Se está implementando actualmente en RASA.

In: tipo de bloqueo, eventos en el grafo

Out: generación del diálogo



NOTA: Los **puntos de control** del diálogo deberían ser, como mínimo, que el enlace `is_blocking` se elimine. Esto es, el agente `socialNavigation` debe monitorizar esto.

Otra información de interés para el diálogo: nombre de la persona, hacia dónde va el robot (si es una terapia...), género...

Durante el diálogo pueden cambiar las prioridades de la misión, por ejemplo, en el diálogo el robot puede modificar su ruta...

Descripción del caso de uso completo

El robot recibe una nueva misión: moverse a una nueva habitación con un **objetivo A**.

Los agentes insertan los objetos y las personas, se calculan los espacios personales y las posibles interacciones. El robot planifica su ruta social.

El robot inicia su movimiento.

Si no hay bloqueo en ningún momento, el robot alcanza su destino.

Si durante la navegación, la persona se mueve y bloquea el camino, el agente de navegación no puede llegar a destino y genera el enlace **'is_blocking'** de la persona al robot. Además, se añade el enlace **'blocked'** del robot a sí mismo. El robot cambia su ruta hasta acercarse a la persona.

Una vez cerca de la persona, se monitoriza si la persona está de frente (enlace **'front'**) o de espaldas. Si está de frente, comienza a hablar con el diálogo específico para este caso de uso. (askForPermission)

Si la persona no está de frente², comienza un diálogo para llamar la atención de la persona. (takeTheAttention)

Si se da la vuelta, comienza el diálogo específico.

Si no se da la vuelta, el robot lo intenta por segunda vez (el mismo diálogo para llamar la atención). y si no, el robot modifica igualmente el enlace **'front'** como si el diálogo hubiera sido efectivo.

Si el diálogo específico comienza, el conversacional debe llevar a cabo este control mediante el agente y su implementación en RASA. Actualizar los enlaces adecuados (enlace **is_interacting**). El agente debe monitorizar el estado del enlace **'blocked'** (que debe ser observado por el agente de navegación cada cierto tiempo, por si este cambia antes de tiempo, cambiar el curso del diálogo).

El diálogo puede ser más personalizado si usa datos de la persona o de la misión que tiene asignada.

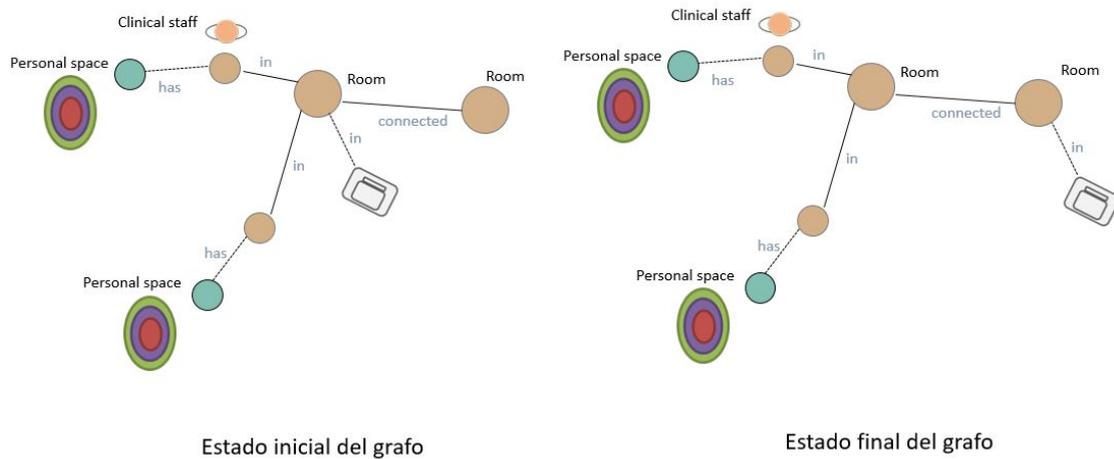
Si la persona se niega, por cualquier motivo, a moverse, el enlace **'block_renewed'**, el robot se retira, se activa tras un cierto tiempo el enlace **'block'** y el robot reinicia de nuevo el intento.

NOTA para pensar: Los objetivos del robot deben tener cierta escala de prioridad. A > B ... > K. Si en el diálogo la persona le pide un cambio de misión, se podrá siempre que esta prioridad sea mayor

² el atributo 'front' o similar no lo tenemos, podría ser interesante para ser más social

DSR para el caso de uso 1. Nodos y enlaces

La siguiente figura muestra el caso de uso 1. Se detallan a continuación cómo deben ser los nodos y enlaces del DSR al inicio de la navegación y al fin de la misma. En este caso el robot navega de una habitación a otra.

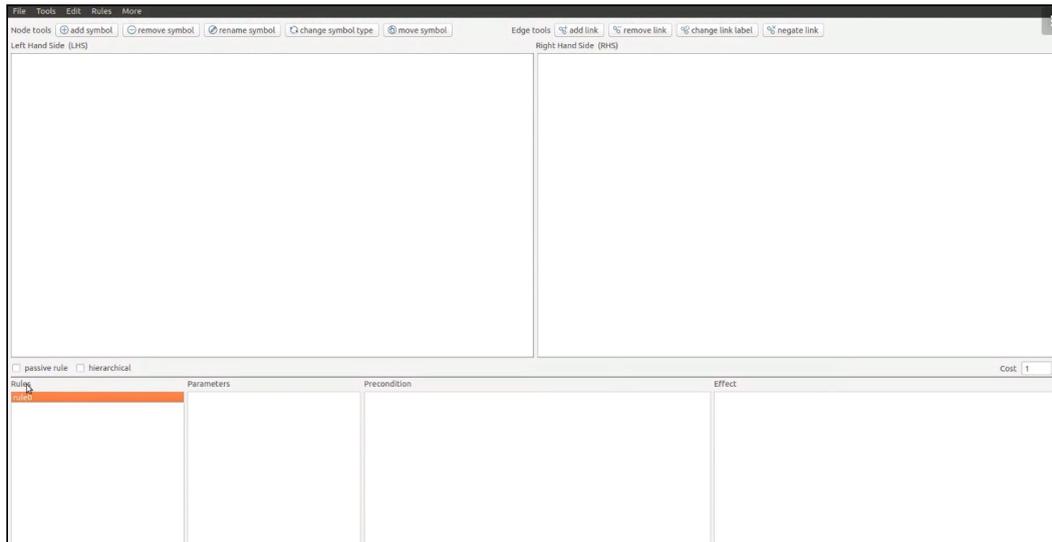


Planificación con AGM

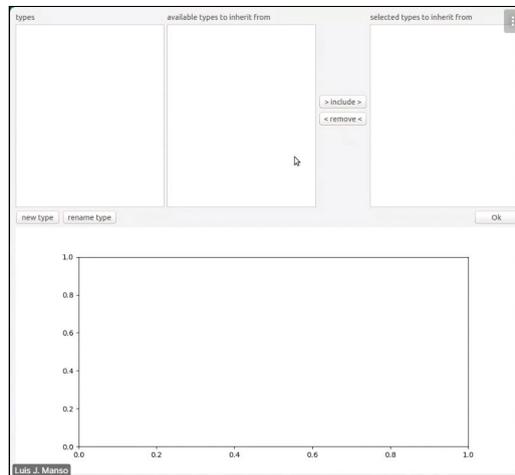
Herramientas

Editor de dominios de planificación basado en el lenguaje AGGL

El programa nos permite la edición de acciones basadas en reglas de una forma gráfica e intuitiva. La apariencia del programa es la siguiente:



Cada acción del robot es una regla atómica. La definición de reglas se realiza gráficamente con la barra superior. Se incluyen funciones básicas para crear, editar y borrar símbolos y enlaces. Los tipos se describen previamente en el menú edit-type hierarchy.



Definición de reglas. Modelos de conocimiento del robot

Reglas básicas y genéricas para el caso de uso 1

1. Control de la batería del robot (rule: battery_on_off).

Agente involucrado **batteryAgent**

Si el robot se queda sin batería (por debajo del 25%), es importante que el robot cambie su ruta y se dirija a la base para la carga.

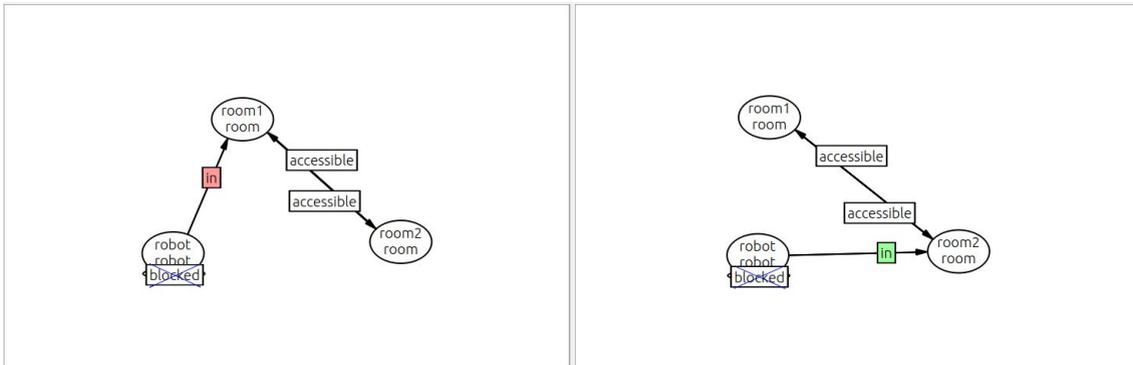


batteryAgent: agente encargado de enviar al robot a la base de carga y monitorizar el estado de la batería del robot.

2. Navegar a otra habitación (rule: changeRoom).

Agente involucrado: **socialNavigationAgent**

El robot debe navegar socialmente a otra habitación que esté accesible con la habitación inicial.

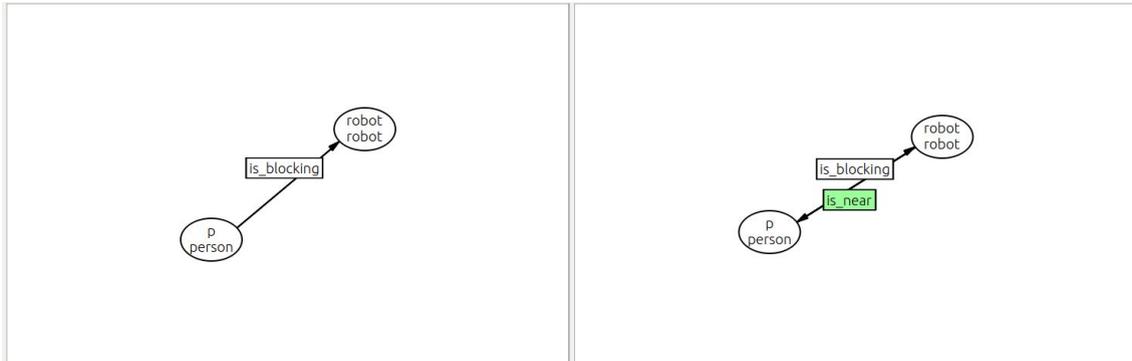


SocialNavigationAgent: agente encargado de realizar una navegación social en el entorno.

3. Navegar hacia una persona (rule:goToPerson).

Agente involucrado: **socialNavigationAgent**

El robot debe navegar hacia una persona hasta llegar a estar cerca de ella. Normalmente esta funcionalidad del navegador será necesaria en aquellos casos en los que el robot precise interactuar con la persona en el entorno.



Precondición:

```
(forall
  otherp:person
  (when
    (not (= p otherp))
    (not (is_blocking otherp robot))
  )
)
```

La precondición se pone para que solo se ejecute esta regla cuando haya una sola persona bloqueando al robot.

SocialNavigationAgent: agente encargado de realizar una navegación social en el entorno.

4. Llamar la atención de la persona, antes de interactuar (rule:takeTheAttention).



Agentes involucrados:

faceinfrontAgent monitoriza cuando la persona está de frente al robot. Se activa la atención cuando está cerca.

Agente HRIAgent lanza un diálogo específico para llamar la atención.

Tras el diálogo puede haber cambiado el enlace 'front' o no. Tras un timeout, se activará el enlace 'front' como si el diálogo hubiera sido efectivo.

5. Pedir permiso a una persona (rule: askForIndividualPermission)



Precondición:

```
(forall
  otherp:person
  (when
    (not (= p otherp))
    (not (is_blocking otherp robot))
  )
)
```

La precondición se pone para que solo se ejecute esta regla cuando haya una sola persona bloqueando al robot.

Agentes involucrados:

SocialNavigationAgent: monitoriza el estado del enlace. En caso de que la persona se retire, se debe eliminar el bloqueo.

conversationalAgent: lanza un diálogo específico para eliminar el enlace de camino bloqueado. El diálogo se realiza a través de RASA y tendrá en cuenta información del grafo disponible. Por ejemplo, si se conoce el nombre de la persona, o la categoría (senior, profesional clínico, cuidador...), el diálogo será más personalizado. En caso de que el bloqueo se elimine antes de acabar el diálogo, el agente debe adaptar el flujo de la conversación de forma natural.

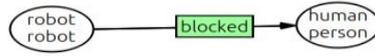
En caso de que el diálogo acabe y el bloqueo no se elimine, esto es, la persona por algún motivo no se retira de su posición y por tanto el robot no alcanza el destino, tras un timeout se crea un nuevo enlace en el grafo (blocked_renewed)

6. La persona no se retira y elimina el bloqueo (rule: path_blocked_renewed)

Agentes involucrados:

HRIAgent lanza un diálogo específico para retirarse tras una conversación anterior fallida.

SocialNavigationAgent: se retira a una posición libre alejándose de la persona. Al finalizar el movimiento, activa de nuevo el enlace 'blocked'



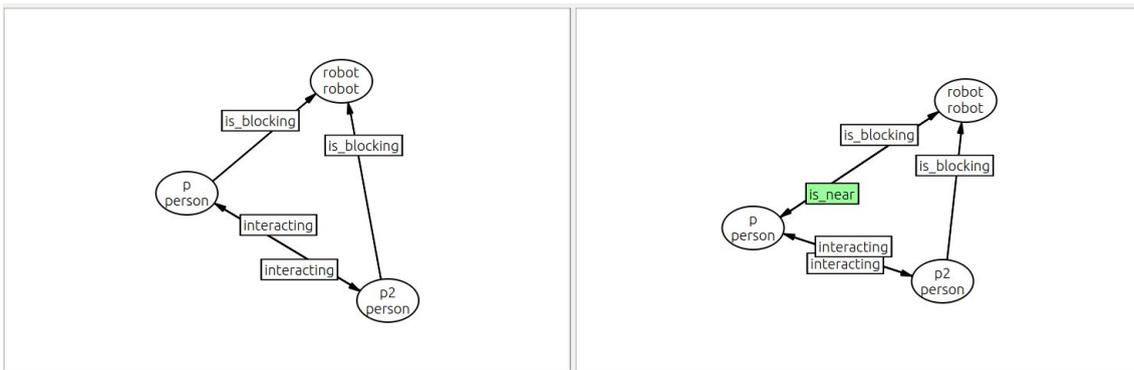
Reglas básicas y genéricas para el caso de uso 2

Las reglas definidas para el caso de uso 2 son idénticas a las definidas para el caso 1 con dos excepciones:

1. Navegar hacia un grupo de personas (rule:goToGroupOfPeople).

Agente involucrado: **socialNavigationAgent**

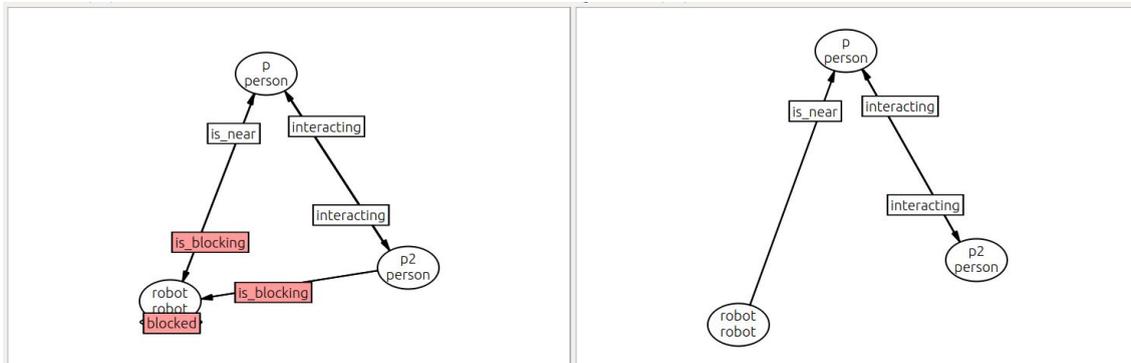
El robot debe navegar hacia la persona más cercana del grupo hasta estar cerca de ella.



Se debe cumplir que las dos personas estén interactuando, a parte de que las dos bloqueen al robot. Este caso se cumple para dos o más personas.

SocialNavigationAgent: agente encargado de realizar una navegación social en el entorno.

2. Pedir permiso a un grupo de personas (rule: askForGroupalPermission)

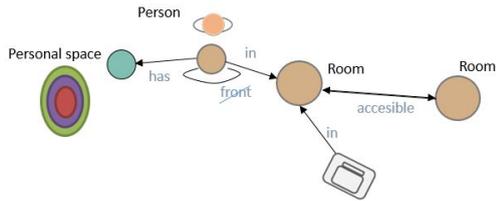


Agentes involucrados:

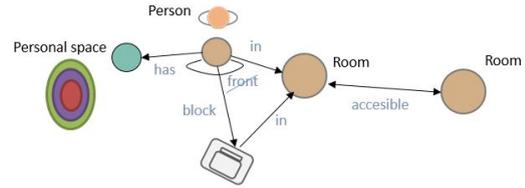
SocialNavigationAgent: monitoriza el estado del enlace. En caso de que una de las personas se mueva y el robot tenga espacio para pasar se elimina el enlace.

conversationalAgent: lanza un diálogo específico para eliminar el enlace de camino bloqueado. El diálogo se realiza a través de RASA y tendrá en cuenta información del grafo disponible. Por ejemplo, si se conoce el nombre de la persona, o la categoría (senior, profesional clínico, cuidador...), el diálogo será más personalizado. En caso de que el bloqueo se elimine antes de acabar el diálogo, el agente debe adaptar el flujo de la conversación de forma natural.

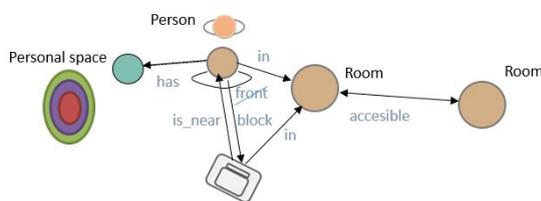
Evolución del grafo durante el caso de uso



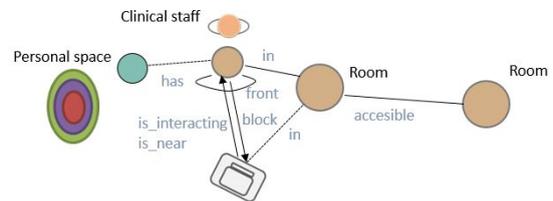
Estado inicial del grafo



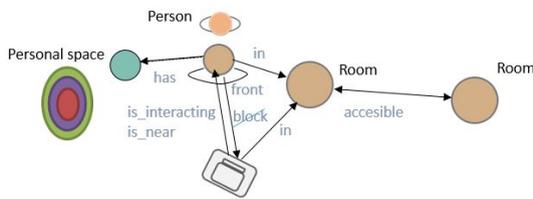
El robot comienza su navegación y se encuentra un bloqueo



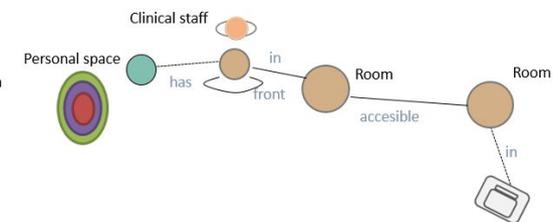
Navegar hasta situarse cerca de la persona



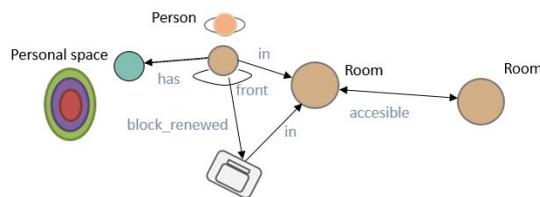
Se inicia el diálogo para llamar la atención de la persona



Se inicia el diálogo para intentar liberar el bloqueo con éxito



Se reanuda la navegación hacia el destino



El robot se retira si no hay éxito en el diálogo de desbloqueo