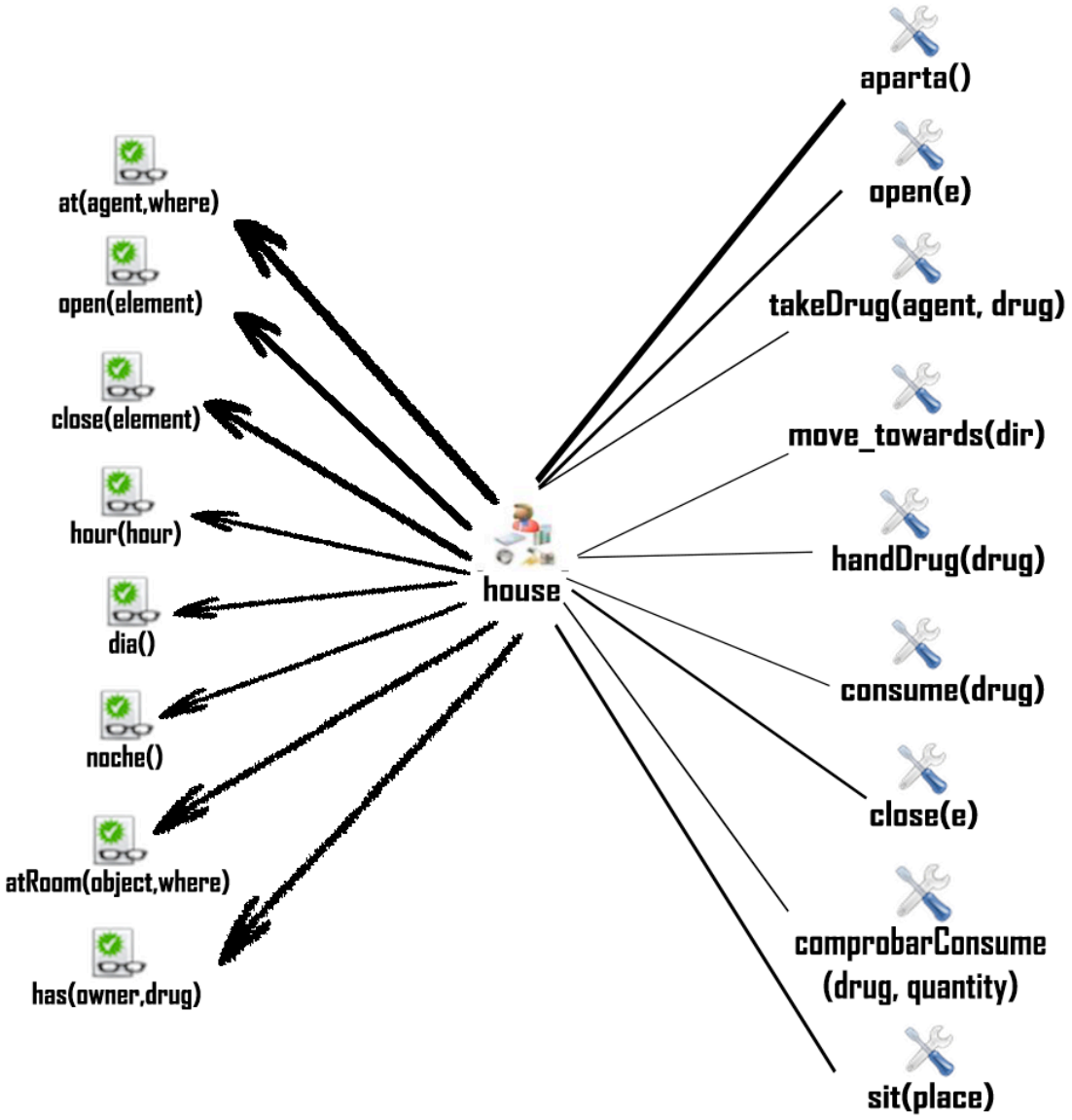
Memoria Sistemas Inteligentes

Raúl Blanco Garrido
Erik Figueiral Alonso
Manuel Lechuga Meirás
Erik Pereira Fernández

Sistema Multiagente conformado por los agentes: Robot y Owner,el primero actuará como enfermero proporcionando medicinas al segundo, que es el propietario de la casa.Ambos agentes podrán moverse libremente,interactuar entre ellos y con el entorno.
Para lograr implementar esta funcionalidad utilizaremos un modelo vista controlador utilizando AgentSpeak implementado a través de Jason.

Comenzaremos describiendo el entorno con el que convivirán los agentes

Diagrama del entorno



Acciones Externas:

- ***Aparta():*** Tras la comunicación de agentes,uno de ellos se apartara de forma aleatoria.
- ***Open(e):*** Permite abrir el elemento.
- ***TakeDrug(agent, drug):*** El agente,obtiene la droga del cabinet.
- ***Move_towards(dir):*** Permite a los agentes moverse con un algoritmo A*. Tiene dos modos de funcionamiento:
    - Se tienen en cuenta los agentes.
    - No se tienen en cuenta los agentes.

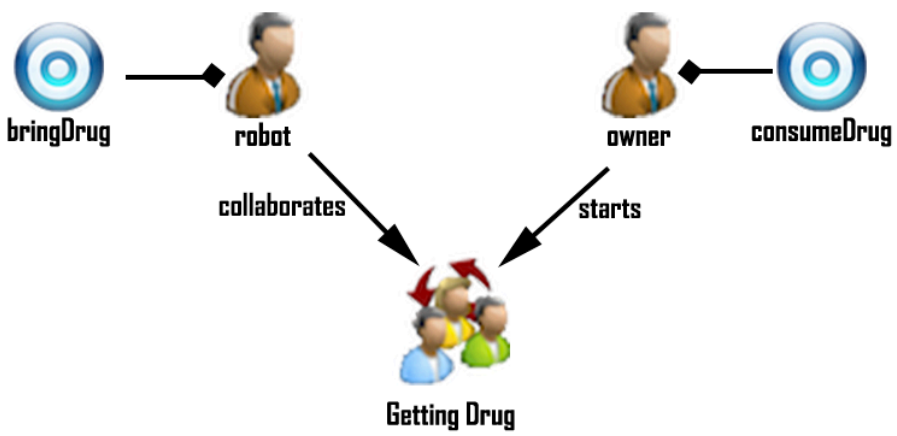
  En el caso de que falle el primer modo de funcionamiento, se lanzará el segundo. De esta manera forzamos que la colisión entre agentes sólo pueda producirse en casillas adyacentes,
- ***HandDrug(drug):*** Se obtiene la droga y se reduce la cantidad en el cabinet.
- ***Consume(drug):*** Se consume la droga.
- ***Close(e)***:Permite cerrar el elemento.
- ***ComprobarConsume(drug,quantity):*** Con un conocimiento previo de la cantidad existente de dicho tipo de droga,se comprueba si se ha producido una consumición
- ***Sit(place):*** Permite sentarse.

Percepciones:

- ***at(agent,where)***: Informa de la posición actual del agente
- ***open(element)***: Percepción de que dicho elemento está abierto.
- ***close(element)***: Percepción de que dicho elemento está cerrado.
- ***hour(hora)***: Hora actual del sistema.
- ***dia()***: Percepción de que es de día.
- ***noche()***: Percepción de que es de noche.
- ***atRoom(object,where):*** Percepción donde se encuentra un objeto.
- ***has(owner,drug)***: El agente owner,tiene dicha droga en la mano.

Vamos a identificar los objetivos generales del sistemas SMA que desarrollaremos.
Recordemos: Tenemos 2 agentes, robot y owner con objetivos distintos,respectivamente **bringDrug** y **consumeDrug**.

Diagrama de organización



Los objetivos ideales en nuestro Sistema será que el robot entregue en la Pauta indicada las medicinas al Owner, para ello en la hora indicada tendrá que **entregarMedicina(L)** ,es decir las medicinas que le tocan en esa hora.
Para lograrlo tendra que **bringDrug(L)** ,que se descompone en **goAtCabinet**, se cogen las medicinas **takeDrug**,una vez en nuestra posesión tenemos que ir al owner **goAtOwner** y entregarselas **handDrug**, cuando las tenga en la mano **hasDrug** las podrá consumir que se traduce en el objetivo consume.

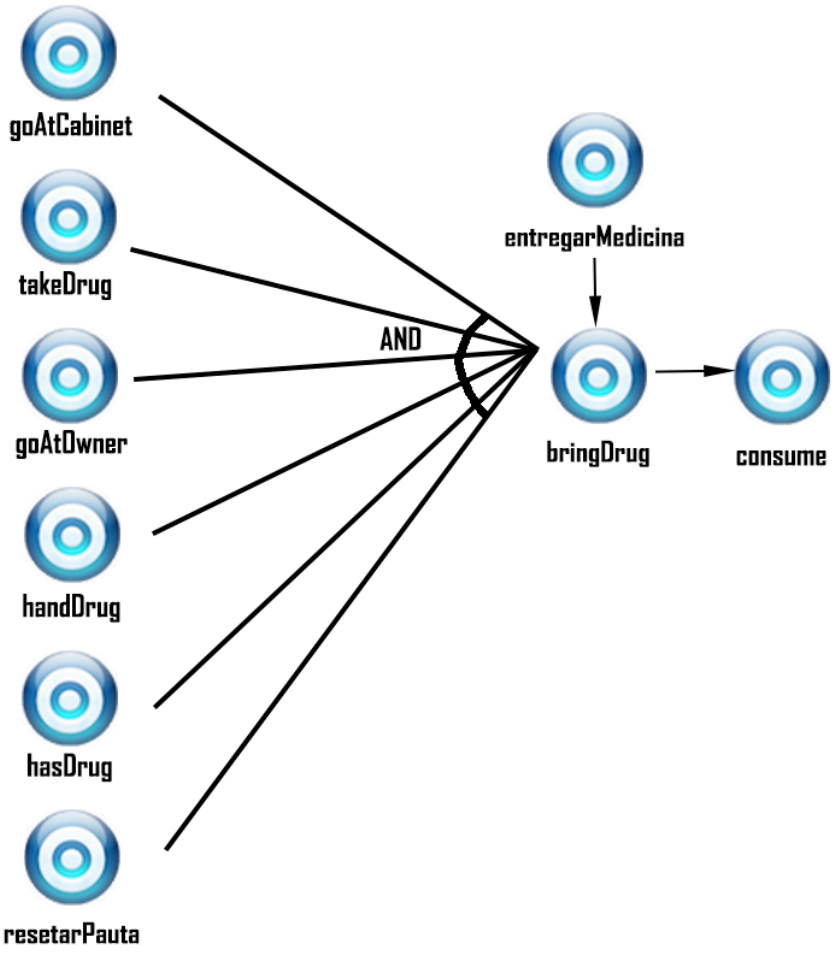


Diagrama de objetivos ideal

Creamos un diagrama de interacción apoyándonos en el diagrama de organización.

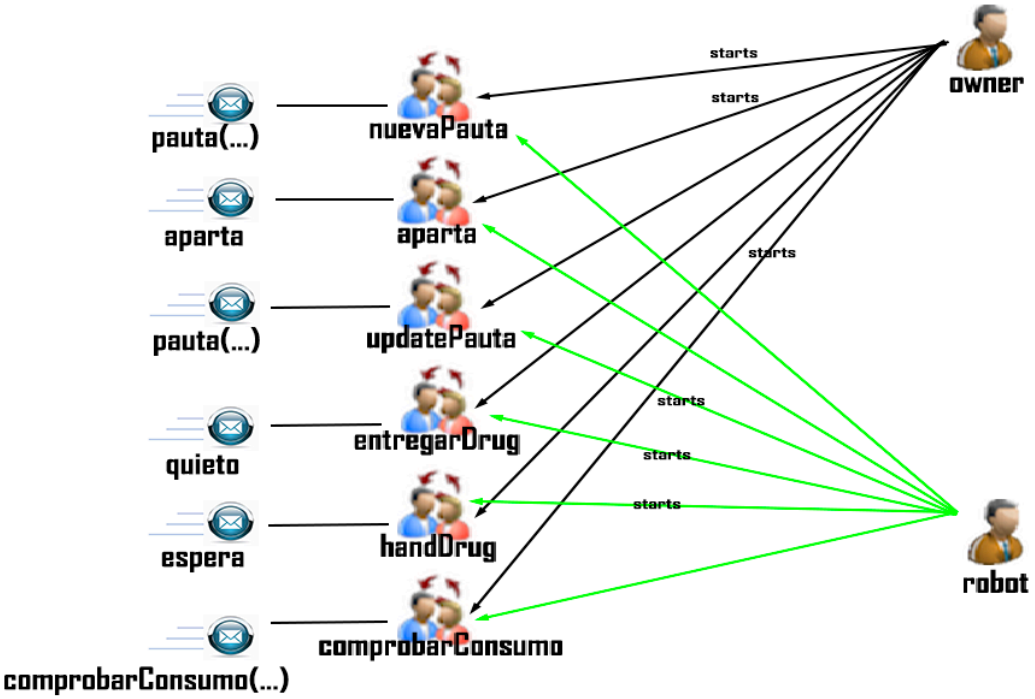


Diagrama de iteracción

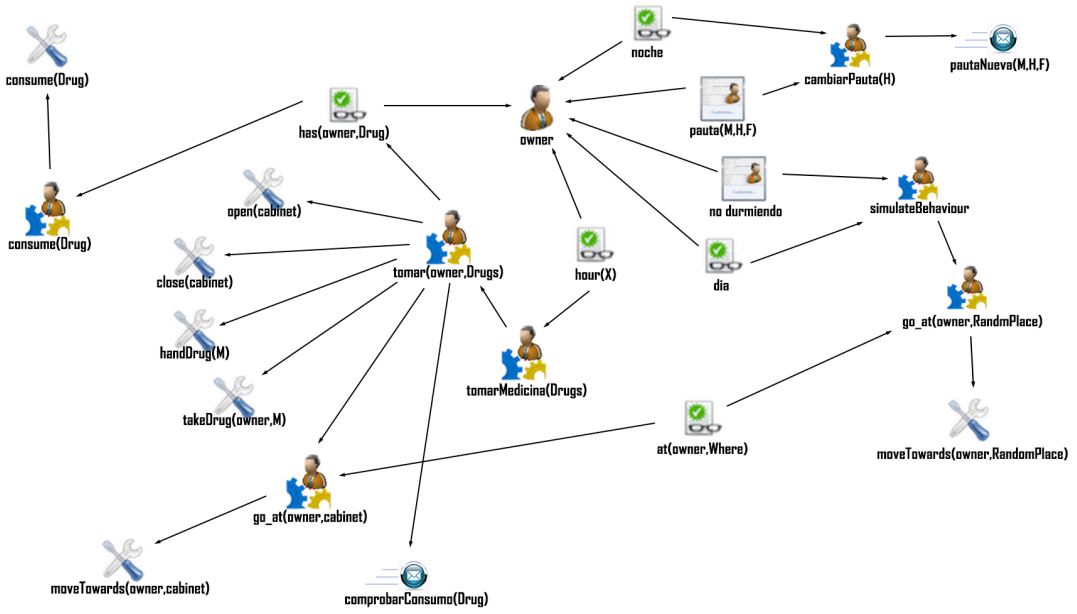


Diagrama de tareas del owner

El owner, tan pronto se inicia y cuando llegan la noche calcula las pautas para cada medicina mediante el plan cambiarPauta(H), comunicándoselas al robot mediante un mensaje.

Cuando se hace de día el owner tiende a simular el comportamiento, esto es ir aleatoriamente a ciertos sitios de la casa mediante el plan simulateBehaviour, mediante el cuál hace varios go_at a sitios random de la casa.

Cuando llega la hora de tomar la medicina, que se comprueba mediante la percepción de hora y las pautas establecidas, tiene una probabilidad del 20% de ir al armario a por las medicinas mediante el plan tomarMedicina(Drugs), el cual suspende las intenciones de simular el comportamiento, y llama al plan tomar(owner,Drugs) que realiza las acciones necesarias para tomar la medicina. Una vez toma la medicina le envia al robot el mensaje comprobarConsumo(Drug), para que este compruebe si realmente ha cogido la medicina que ha dicho.

Una vez el owner tiene alguna medicina en las manos, lo sabe con la creencia de has(owner,Drug), se ejecuta el plan de consume(Drugs).

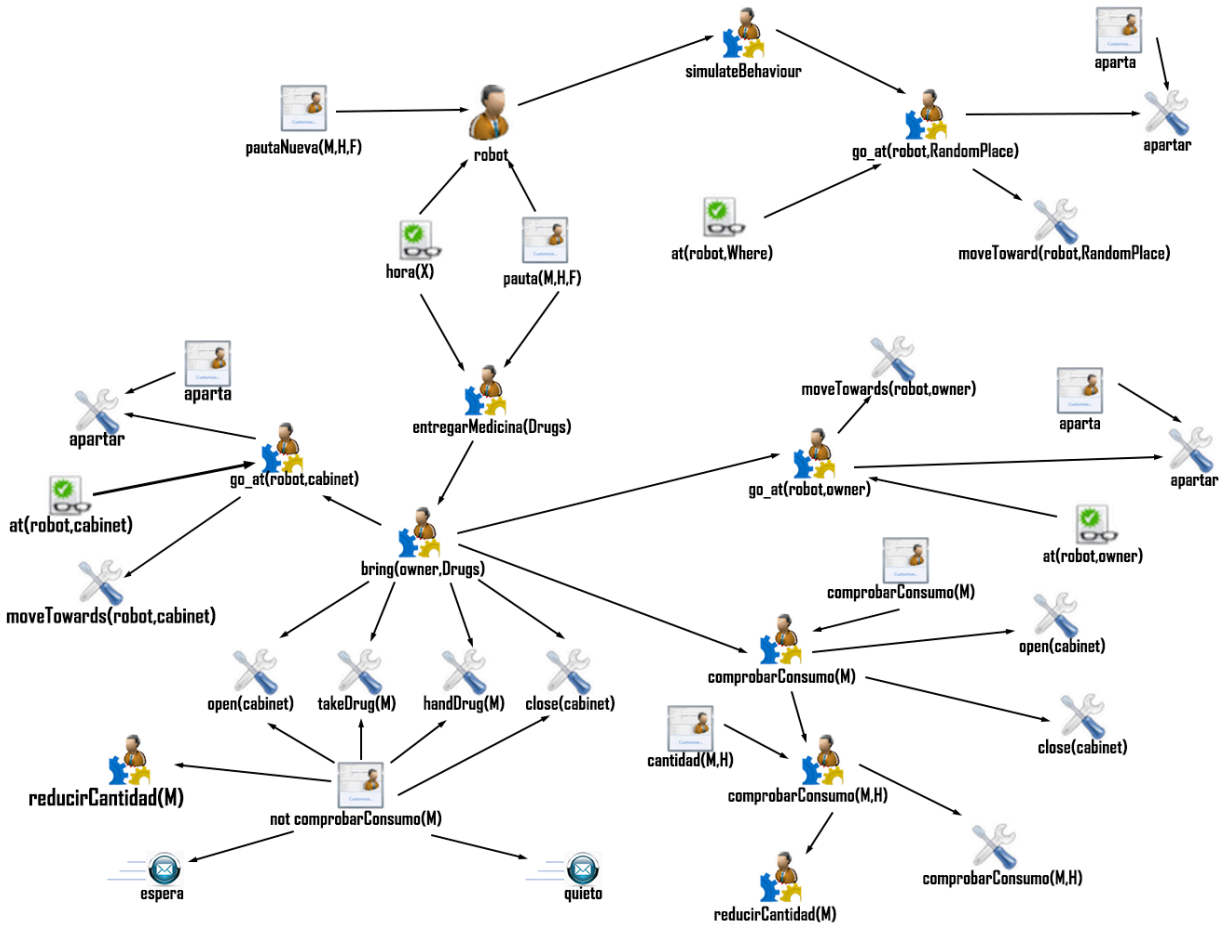


Diagrama de tareas de robot

El robot tiende a simular el comportamiento, esto es ir aleatoriamente a ciertos sitios de la casa mediante el plan simulateBehaviour, mediante el cuál hace varios go_at a sitios random de la casa, tiene que tener cuidado con el owner, en caso de que deba de apartarse.

Cuando llega la hora de la medicina, que lo sabe mediante la creencia de pauta y la percepción de hora, ejecuta el plan de entregarMedicina(Drugs), que se encarga de llevar la medicina al owner. Este plan llama a su vez al plan bring(owner, Drugs) que se descompone de la siguiente forma:
En primer lugar el robot va al armario de las medicinas mediante un go_at.
En segundo lugar realiza las acciones necesarias para coger la medicina, y reducir la cantidad de dicha medicina mediante el plan reducirCantidad(Drug), siempre que el owner no le haya dicho ya que se las ha tomado mediante el mensaje comprobarConsumo, así mismo envía el mensaje de quieto y espera al owner para que no vaya al armario.
Acto seguido va al owner y le entrega sus medicinas.

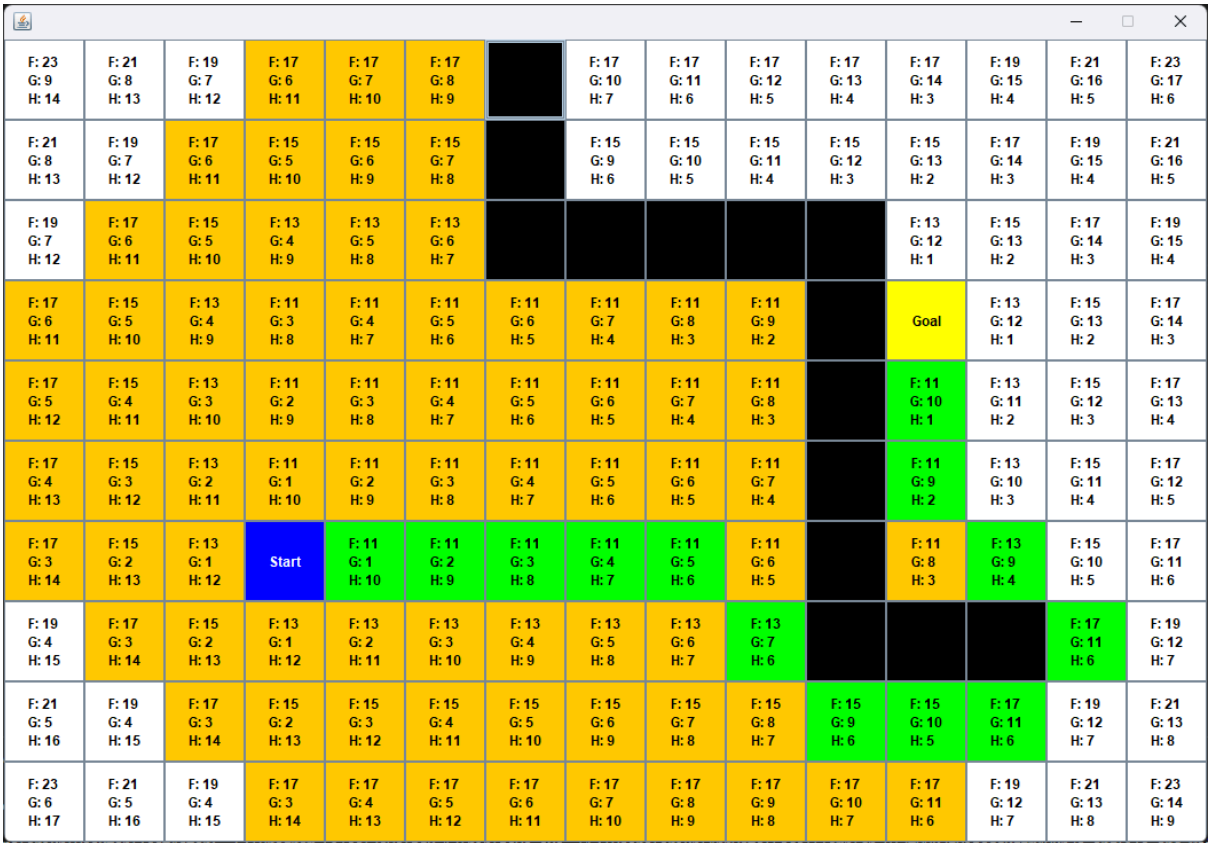
En caso de que el owner haya llegado antes al armario, el robot se encarga de comprobar que ha cogido las medicinas mediante el plan comprobarConsumo(Drug), y también se encarga de reducir la cantidad de medicinas disponible.

El movimiento en el espacio

Para conseguir que nuestros agentes puedan moverse libremente por el espacio hemos recurrido a realizar una implementación de un algoritmo A* en el que el movimiento puede realizarse tanto de manera horizontal como diagonal. El algoritmo A* crea una especie de rejilla del tamaño de nuestra casa y mapea las casillas a las que no se puede acceder, marcándose como sólidas. A las casillas a las que puede acceder nuestro agente le realiza una serie de cálculos heurísticos. Podemos distinguir 3 tipos de coste en las casillas:

- Coste G: Es la distancia Manhattan al nodo origen.
- Coste H: Es la distancia Manhattan al nodo destino.
- Coste F: Es la suma de ambas distancias.

El trabajo del algoritmo es, entonces, calcular estas distancias, y recorrer el camino desde el nodo objetivo (Localización objetivo) hasta el nodo origen (Localización origen) observando los 8 nodos adyacentes al que está en observación y trazando el camino en base al que tiene un menor coste F, en caso de que el coste F de ambos nodos sea igual, se busca moverse al que esté más cerca del nodo origen, esto es, el que tenga un menor Coste G.



Prueba del algoritmo desarrollado

Como se puede observar se examina el camino hasta llegar a destino y se coge la ruta óptima en cada momento.

En nuestro caso en primer lugar el algoritmo tiene en cuenta a los agentes, es decir, si un agente bloquea el único camino existente, se devuelve un path nulo, entonces se lanza otra estrategia, no tener en cuenta al agente y realizar las acciones oportunas(apartarse) cuando se choque con él.
Si bien el algoritmo funciona perfectamente siempre que exista un path entre origen y destino, ¿Qué pasa cuando los dos se encuentran en el pasillo en direcciones opuestas?
En nuestra solución el owner envía un mensaje al robot pidiéndole que se aparte, esto es, que se mueva en una posición aleatoria válida en sus adyacentes de forma que eventualmente ambos consiguen pasar y llegar a su destino.

Tanto el owner cómo el robot pueden moverse de forma aleatoria entre una serie de puntos en el mapa. Debemos de tener en cuenta que el owner sólo se moverá cuando no esté durmiendo, esto es, cuando la hora no esté entre las 00:00 y las 8:00. Cuando llega la hora de tomar las medicinas y deciden que deben ir a por ellas, ambos dropean la intención en curso para ir a realizar la acción de ir a por las medicinas ya que es más importante.

Descompondremos ahora las acciones de nuestro proyecto.

Para este proyecto hemos diseñado una clase Calendar de Java que incorpora un Slider con la hora que es, pudiendo ser cambiada manualmente (Obviamente no pasar 15 horas seguidas porque como es normal el programa no va a soportarlo). Las pautas de las medicinas son generadas aleatoriamente por el owner una vez se inicia el programa y estas son transmitidas al robot mediante un send tell. Una vez llega la hora(ambos agentes saben que hora es ya que el entorno se lo updatea) el robot va a dirigirse siempre hacia el armario de las medicinas, el owner tiene una probabilidad del 20% (si no está durmiendo) de ir allí también.

En caso de que ambos agentes quieran ir al armario, se produce una condición de carrera, que gana el que primero llegue al armario. Entonces se nos plantean 2 situaciones:

- El robot llega primero al armario:
    - Le dice a owner que se pare, para esto le manda un send tell de quieto.
    - Hace la acción de bring(owner,drugs).
- El owner llega primero al armario:
    - El owner realiza su acción de tomarMedicina e informa al robot de que ha tomado las medicinas, cosa que este comprueba cuando le llega un mensaje de tipo tell comprobarConsumo(Medicina).

Acción de llevar las medicinas del robot:
En primer lugar cuando cambia la hora se busca en la base de creencias del robot alguna medicina (o medicinas) que unifiquen con esa hora. De existir alguna se dropea la acción que esté realizando el robot en ese momento y se va al armario por las medicinas.

Una vez llega al armario, y si el owner no está ya en el, le dice al owner que se esté quieto y procede a:

1. Abrir el armario.
2. Para cada medicina, la coge, reduce su cantidad en 1 unidad e informa de que ha cogido esa medicina.
3. Cierra el armario.

Una vez tiene las medicinas, se dirige hacia el owner, y por cada una de las medicinas se la da y después recalcula la siguiente pauta de toma de esa medicina, sumando la hora a la que la ha tomado la frecuencia con la que debe de hacerlo, teniendo en cuenta que esto no puede ser superior a 24, una vez obtiene la nueva pauta se la transmite al owner.

Acción de coger las medicinas del owner:
En primer lugar cuando cambia la hora se busca en la base de creencias del owner alguna medicina (o medicinas) que unifiquen con esa hora. De existir alguna y que el owner decida ir a por ellas, se dropea la acción que esté realizando el robot en ese momento y se va al armario por las medicinas.
Si llega al armario satisfactoriamente, esto es, si el robot no le ha dicho que se pare y el robot no está ya en el armario hace lo siguiente:

1. Abre el armario.
2. Coge cada medicina.
3. Indica al robot que compruebe el consumo de dicha medicina.

Cuando el owner tiene alguna medicina en su poder, esto es, no está vacía una Lista de Java que almacena las medicinas que tiene en ese momento en sus manos, se invoca el método consume, que en lugar de hacer sips de medicina considera que la ha tomado al pasar unos segundos. Esto se realiza **SIEMPRE** que el owner tenga una medicina en su mano, venga de la fuente que venga.

Cuándo se hace de noche (00:00) el owner tendrá la percepción de que es de noche, con lo que se irá a una de las tres camas aleatoriamente, y reiniciará las pautas de forma aleatoria,es decir, le comunicará al robot las nuevas pautas que se deben entregar las medicinas.En el caso de que tenga que tomar una medicina durante la noche,se la entregará el robot mientras él descansa.

Semana 5

Objetivos de la Primera Fase del Proyecto

En esta primera parte hay que realizar una modificación al ejemplo de robot domestico que ya se ha presentado, para transformarlo en un proyecto de un agente asistencial doméstico. En esta primera versión tendremos solo dos agentes: el agente owner y un agente robot (enfermera).

- El agente owner tiene necesidad de tomar sus medicaciones pautadas (deberéis elegir al menos 5 medicamentos y la pauta de toma de cada uno de ellas) El agente owner puede moverse libremente por la casa y descansar en zonas adecuadas para ello (silla, sofá, cama, ...)

  Hemos hecho que el owner tenga 5 conocimiento de medicamentos,donde se establece el nombre,hora y frecuencia del mismo.

  Para simular su movimiento hemos establecido 3 comportamientos según la probabilidad,ir hacia un objeto/elemento de la casa,sentarse o irse a dormir.
- La medicación puede ser proporcionada por el agente robot o por el propio agente owner. Para que el robot pueda servir los medicamentos al owner, este debe indicar la pauta de las medicaciones al robot nada más crearse.

  El agente robot comprobará en los cambios de hora si tiene que proporcionar una pauta,en dicho caso se las proporcionará que recibirá las pautas a la creación y cuando sean modificadas a la noche por el owner.El owner tiene una probabilidad pequeña de ir a por ellas.
- En caso de haber tomado la medicación, el owner debe indicarlo al robot.

El robot comprueba que el owner ha tomado los medicamentos si este los ha cogido del cabinet,en caso contrario que el se lo proporcione el es cociente de haberselo entregado en sus manos y de su consumición en una lista en el env.

- Con el paso del tiempo, el owner modificará la pauta de sus medicaciones y añadirá y/o eliminará algunas de ellas. Las medicaciones deberán guardarse en sitios adecuados y accesibles tanto para el owner como para el robot.

El owner cuando es la noche se va acostar,en este momento le informa al robot de las nuevas pautas.

- La medicación estará disponible en cantidad siempre suficiente para poder ser administrada.

Hemos asumido que una cantidad de 50 es suficiente para nuestra simulación.

- El agente robot debe poder moverse libremente por la casa sorteando los objetos que en ella se encuentra, de una habitación a otra, buscando al owner para entregarle la medicación cuando proceda.

Ambos agentes presentan un algoritmo muy utilizado y óptimo en path-finding que es A*.

- Si el robot recibe indicación de que alguna medicación se ha tomado, debe comprobar que ha sido así.

El robot comprueba en el cabinet que se ha reducido 1 cantidad de dicho medicamente,en este caso asumirá que el owner se lo ha tomado.

Semana 6

El objetivo de esta semana es conseguir que:

1. Los agentes owner y robot puedan desplazarse por toda la casa evitándose y evitando los obstáculos que puedan aparecer en su camino.

   Algoritmo A* y simulación de comportamiento en ambos agentes.
2. El owner envie al robot los medicamentos y las pautas iniciales.

   Send inicial en un bucle for de su conocimiento de pautas.

NOTA: Fijaos que el punto 1 conlleva que el agente owner deja de ser un mero objeto del entorno, para tener actividad propia de desplazamiento por el mismo y comparta objetivos con el agente robot

Semana 7

Objetivos de la Semana

El objetivo de esta semana es conseguir que:

1. El agente owner vaya al armario de las medicinas (de forma aleatoria) y tome la medicación que le corresponda según la pauta marcada y avise al robot de la medicación tomada (cuidado con posibles colisiones en las horas pautadas)

   Tiene una probabilidad de 0.2 de ir a por las medicinas,avisa al robot de que compruebe su consumición si el llega antes.

2. El agente robot compruebe si efectivamente la medicación se ha tomado.

   Comprueba con su conocimiento de la cantidad que había la última vez que abrío la nevera y comprueba si hay una cantidad de existencias igual a la que sabía que había menos uno.
3. El agente robot entregue las medicinas al owner en las pautas indicadas.

   El robot comprueba en el cambio de horas si tiene que entregar las medicinas,en cuyo caso se las entregará,primero moviendose al cabinet,portando las medicinas necesarias,acercándose al owner y por último se las da en la mano.