

Facultat d'Informàtica de Barcelona  
Grau en Enginyeria Informàtica  
Especialitat de Computació i Enginyeria de computadors

## SID: Pràctica 3 laboratori

*Autors: Dámaso-Pierre De La Cruz Espinasse, Pol Marcet Sardà*

Data: 20/05/2023

<b>Agente Situado.....</b>	<b>3</b>
<b>Agente BDI.....</b>	<b>5</b>
<b>Metodología y repartición de tareas.....</b>	<b>7</b>

# Agente Situado

Para la implementación del agente situado, hemos optado por definir un set de acciones iniciales a realizar al situar al agente en el mundo, siendo estas:

## -Registrarse en el DirectoryFacilitator

```
ServiceDescriptor: Register  
Name("polydama-situated");  
Type("polydama-BDIagent-situated");  
Ontologies("polydama-mapstate");  
Languages(FIPANames.ContentLanguage.FIPA_SL);
```

## -Buscar a su agente deliberativo asignado en el DirectoryFacilitator

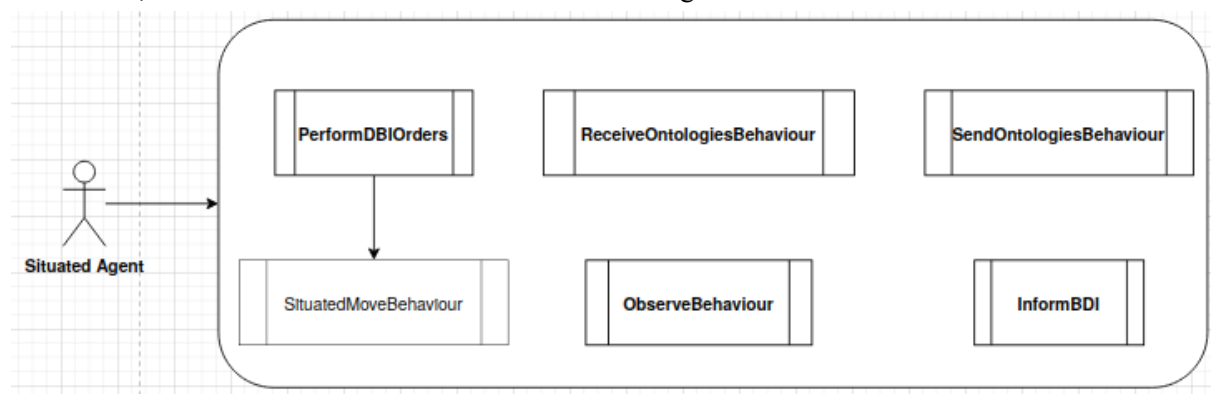
```
ServiceDescriptor: Search  
Type("BDIagent");  
Ontologies("polydama-mapstate");
```

## -Buscar en el DF al resto de agentes registrados

```
ServiceDescriptor: Search  
Ontologies("polydama-mapstate");
```

Respecto a la representación del conocimiento y del entorno, seguimos utilizando la misma ontología que en la entrega anterior, con el añadido de una ObjectProperty funcional “IntendsToWalkTo” [Dominio: Agents; Range: Cells] pero para el caso del agente situado, no se hacen inferencias [OntModelSpec.OWL\_MEM].

Finalmente, describimos brevemente los behaviours del agente situado:



## PerformBDIOrders:

Al recibir la performativa “REQUEST” y protocolo “Order” del agente BDI, el agente situado verifica que en la ontología recibida con la ubicación a la que se desea mover marcada por la propiedad “IntendsToWalkTo” es accesible, es decir, existe un camino para llegar a dicha casilla. En caso de conocer un camino se responde con la performativa “AGREE”, de lo contrario se contesta con “REFUSE”, dejando a JADE generar el mensaje a partir de una excepción.

Una vez finalizada la orden recibida, se informa al agente BDI con el resultado de moverse.

Nótese que se ejecuta “SituatedRouteBehaviour” para realizar el movimiento acordado e informar con el resultado final, ya sea “DONE” o “FAILURE” mediante excepción.

**ReceiveOntologiesBehaviour:**

Comportamiento estándar para recibir ontologías, bajo el protocolo de “SHARE-ONTO” del resto de agentes del mundo para representar su posición y tipo de agente.

**ObserveBehaviour:**

Se observa el entorno desde la nueva posición del agente, añadiendo a la ontología del agente la nueva información percibida.

**SendOntologiesBehaviour:**

Comportamiento estándar para enviar la ontología, bajo el protocolo de “SHARE-ONTO” y con performativa “INFORM” al resto de agentes del mundo.

**InformBDI:**

En caso de haber actualizado algún statement de la ontología desde el último envío, esta se envía únicamente al agente BDI, bajo el protocolo de “Inform” y con performativa “INFORM”.

# Agente BDI

Al ser un agente deliberativo, ajustamos su implementación para que sea basada en Beliefs, Plans and Goals:

Antes de describirlo, es conveniente remarcar que seguimos utilizando la misma ontología que en la entrega anterior, con el ObjectProperty “IntendsToWalkTo” y para el agente deliberativo si que queremos realizar inferencias [OntModelSpec.OWL\_MEM\_MICRO\_RULE\_INF]

## Beliefs

- **iAmRegistered**: Predicado para representar la creencia de estar registrado en el DF.
- **ontology**: Predicado para representar la creencia del conocimiento del mundo.
- **mapRepresentation**: Predicado para representar la creencia de conocimiento del mapa.
- **situatedAgent**: Predicado para representar la creencia de haber encontrado al agente situado.

## Plans

- **registerPlan**: Se registra en el Directory Facilitator y se actualiza el belief “I\_AM\_REGISTERED”

*ServiceDescriptor: Register*

*Name("polydama-Ontology");*

*Type("BDIAgent");*

*Ontologies("polydama-mapstate");*

*Languages(FIPANames.ContentLanguage.FIPA\_SL);*

- **findSituatingPlan**: Se busca en el Directory Facilitator al agente situado, en caso de encontrarlo se actualiza el belief “SITUATED\_AGENT”

*ServiceDescriptor: Search*

*Name("polydama-situated");*

*Type("polydama-BDIAgent-situated");*

- **StayInformedPlan**: Se buscan los mensajes con protocolo “Inform” ya que el mensaje contiene información actualizada sobre el conocimiento observado hasta el momento por el agente situado.

- **requestMovePlan**: Se obtiene la información de los beliefs del agente deliberativo, en caso de haber nodos por visitar, se escoge el más cercano, de no tener nodos abiertos pendientes se busca algún nodo cerrado que no se haya visitado recientemente, y se crea un msg “REQUEST” bajo el protocolo “Order” para darle un destino al agente situado. Finalmente, solo si el agente DBI recibe un “AGREE” seguido de un “INFORM” se marca como visitado el nodo destino de la orden en la representación del conocimiento. En caso de no recibir contestación alguna se establece un timeout de 1 segundo, o recibir algo distinto a un “AGREE”+”INFORM”, se descarta esa orden.

## Goals

- registerGoal: Objetivo para registrarse en el Directory Facilitator
- findSituatingGoal: Objetivo para encontrar a su agente situado en el Directory Facilitator
- stayInformedGoal: Objetivo para mantener la representación del conocimiento actualizada con las observaciones del agente situado.
- exploreGoal: Objetivo para obtener el nuevo nodo que deseamos que el agente situado explore y mandárselo por mensaje.

Para conseguir los objetivos propuestos de manera coherente se agrupan los objetivos en dos sets:

Goals paralelos (llamemosle a este goal: 'plansGoal') para exploreGoal + stayInformedGoal  
Se agrega un set de sequentialGoal al agente deliberativo que consiste en:  
registerGoal + findSituatingGoal + plansGoal

# Metodología y repartición de tareas

Para llevar a cabo esta entrega hemos seguido la misma metodología que hasta ahora, de aprovechar las horas de clase para hacer planteamientos sobre el trabajo a realizar. Se han acordado horas para reunirnos los fines de semana.

- Pol: Agente situado, MapRepresentation, Agente Deliberativo
- Dama: Doc, Code Reviewer, Comunicaciones
- Houda: Ha asistido a una reunión por discord, ha elaborado algún desarrollo sin integrarse en el código entregado y ha trabajado en un estudio sobre el ratio óptimo de nodos.