

## Intel·ligència Artificial 2

### PAC3 — Prova d'Avaluació Continuada

- Per a dubtes i aclariments sobre l'enunciat, adreceu-vos al consultor responsable de la vostra aula.
- Cal lliurar la solució en un únic fitxer comprimit (memòria de la pràctica i fitxers xml dels 3 exercicis). La memòria ha de ser un fitxer PDF . Envieu la solució utilitzant el Registre d'Avaluació Contínua (RAC) .
- El nom del fitxer ha de ser *CognomsNom\_IA2\_PAC3* amb l'extensió *.zip*
- La data límit de lliurament és el 17 de Desembre (a les 24 hores).
- **Raoneu la resposta en tots els exercicis. Les respostes sense justificació no rebran puntuació.**

### Solució. Abastaments 2



L'objectiu d'aquesta PAC és conèixer el funcionament d'un entorn de desenvolupament multi-agent. En concret es treballarà amb l'entorn SeSAM (<http://www.simsesam.de/>). Inicialment es seguirà un exemple guiat per a familiaritzar-se amb l'entorn, seguidament es demana millorar el sistema multi-agent.

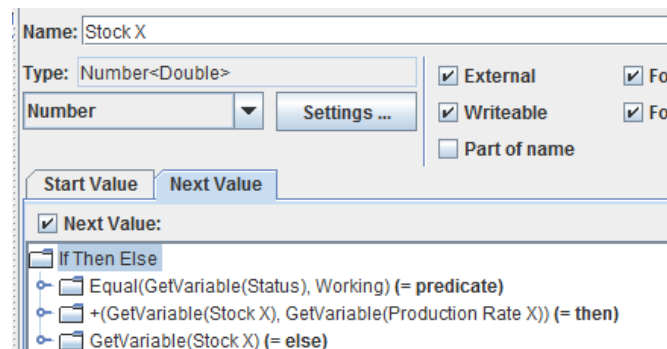
### SeSAM. Tutorial exemple Abastaments (PAC anterior)

L'enunciat es basa en la solució d'una PAC d'altres anys, teniu altres tutorials a <http://130.243.124.21/mediawiki/index.php/TutorialIndex>  
Volem desenvolupar un sistema multi-agent de transport de productes d'una fàbrica a una botiga determinada amb les següents característiques.

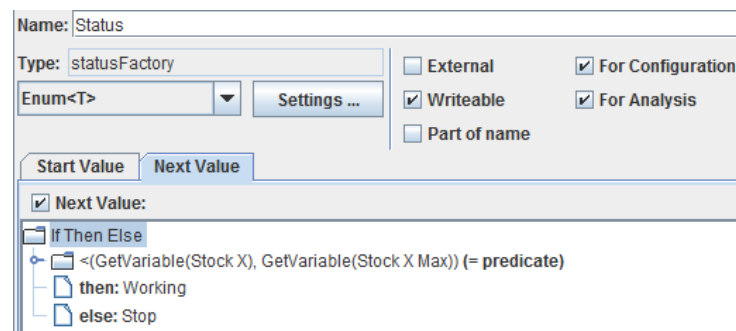
- Agent camió. Agent que transporta el producte X de la fàbrica a la botiga. Té una capacitat màxima de transport de 25 productes. Quan la botiga té espai li porta el producte.
- Botiga. Agent/recurs que ven el producte X rebut de fàbrica. Té un nivell de ventes de 10 productes x hora (hora=iteració) i una capacitat d'emmagatzematge de 100 productes. Té un control del número d'articles venuts.
- Fàbrica. Agent/recurs que fabrica el producte X. Té una producció de 10 productes x hora, i una capacitat d'emmagatzematge de 100 productes. Si arriba a la capacitat màxima no pot produir més (no té on posar-ho).

A continuació veurem els diferents passos per a crear el sistema amb SeSam.

1. Obrir el SeSam i crear un nou model buit. New model
2. **Crear un nou recurs (resource):** “Fabrica”. Al clicar a dins s’obre la finestra per definir-ne les seves propietats, ens centrarem en les variables i el motor de raonament (reasoning engine). Les variables poden ser Writeable (que es poden modificar) i/o externes (que altres agents poden consultar). A més es pot definir el valor inicial (Start Value) i el valor que tindrà següent iteració (Next value). Per exemple, la variable stock X va augmentant a cada iteració fins a arribar al stock màxim utilitzant l’opció next value d’aquesta variable, tal i com mostra la figura.



La variable d'estat (status) canviarà a produint (working) o aturat (stop) depenent si s'arriba al stock màxim. Un altre cop s'utilitza l'opció next value, segons la figura següent.



3. **Creem un altre recurs: Botiga (Store).** De manera similar a la fàbrica, conté les variables stock de producte, stock màxim, vendes/hora, estat (venent o parat). El stock es decrementarà a cada iteració (utilitzant next value) semblant al stock de la fàbrica però disminuint en comptes d'augmentar. De manera similar també s'actualitzarà l'estat.

| Variables                                | Interaction elements | Featu |
|--|----------------------|-------|
| ● Stock X Max (Number<Double>, const)    |                      |       |
| ■ Selling Rate X (Number<Double>, const) |                      |       |
| ● Stock X (Number<Double>, write)        |                      |       |
| ■ Status (statusStore, write)            |                      |       |
| ■ total Sales (Number<Double>, write)    |                      |       |

Global Info

Name: total Sales

Type: Number<Double>

Number ▼ Settings ...

☐ External ☒ Force

☒ Writeable ☒ Force

☐ Part of name

☐ Start Value ☒ Next Value

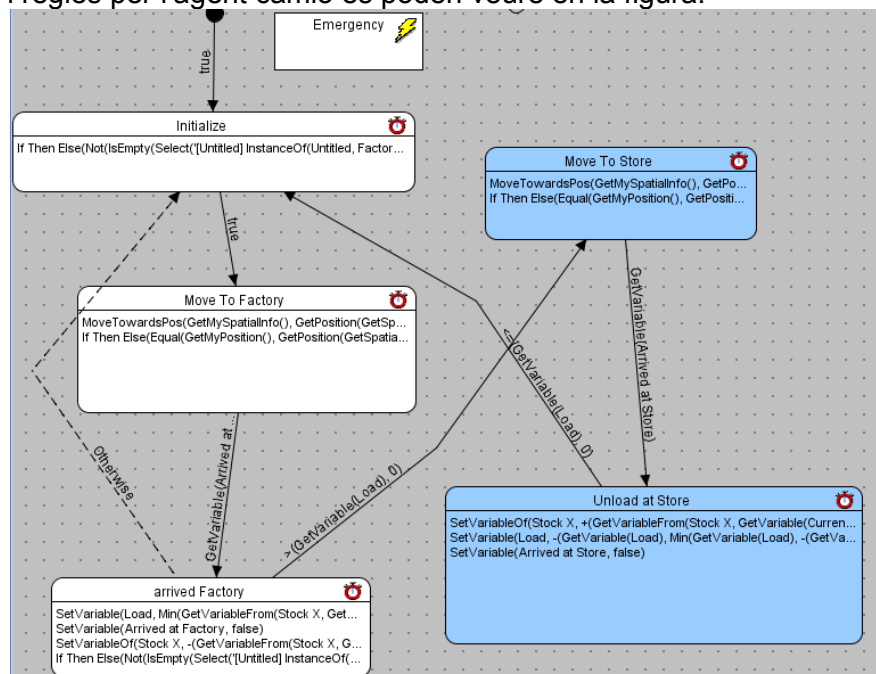
☒ Next Value:

☐ If Then Else

- Equal(GetVariable(Status), selling) (= predicate)
- +(GetVariable(total Sales), GetVariable(Selling Rate X)) (= then)
- GetVariable(total Sales) (= else)

- Max Load (Number<Double>, const)
- Position Of Factory (Position, write)
- Load (Number<Double>, write)
- Arrived at Factory (Boolean, write)
- Currently Observed Factory (SimObject, write)
- Arrived at Store (Boolean, write)
- Currently Observed Store (SimObject, write)
- Position Of Store (Position, write)

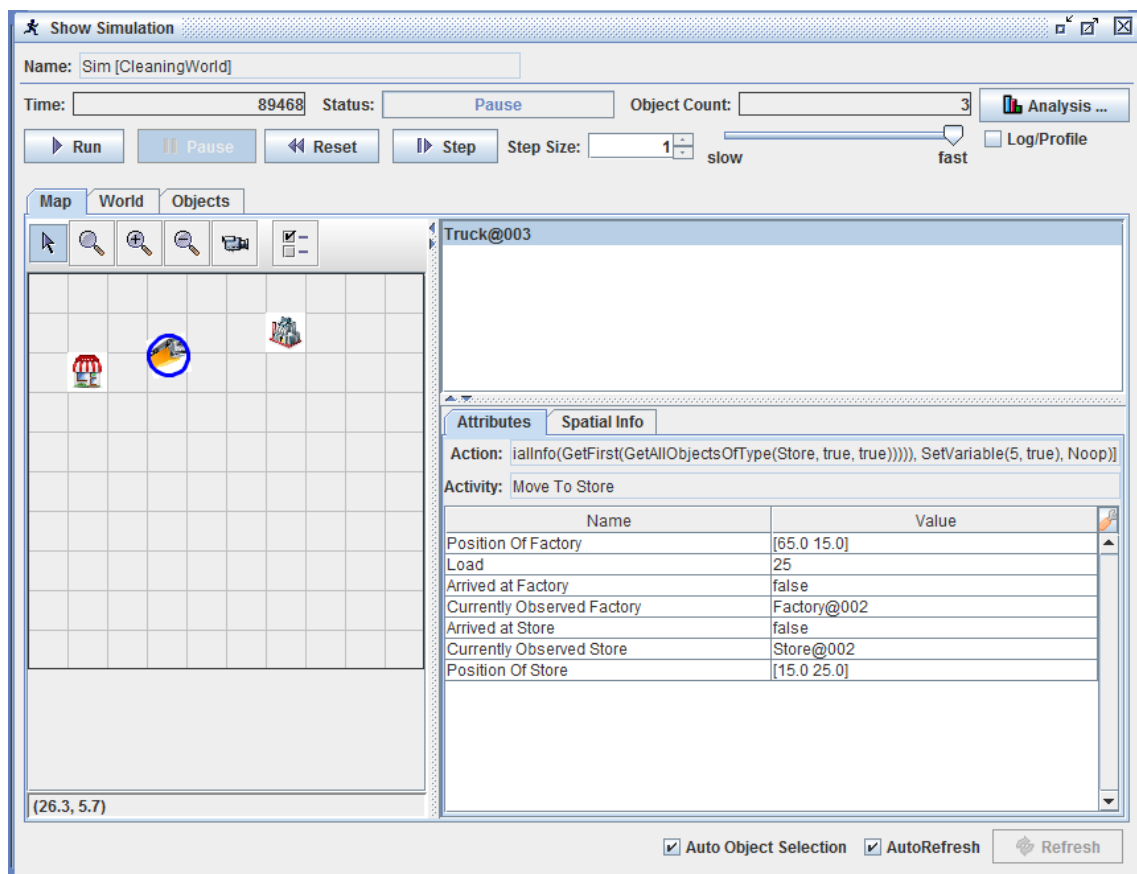
5. **Accions: Activitats i regles.** Tot el procés de funcionament del camió descrit en el punt anterior es troba desenvolupat en diferents accions (*actions*) de l'agent. Cada acció modificarà l'estat o variables dels agents (la seva posició, stocks etc). Per passar de fer una acció a una altra cal definir una sèrie de regles. Les diferents accions i regles per l'agent camió es poden veure en la figura.



Les accions per anar a la botiga o a la fàbrica són similars, bàsicament canviant els noms de les variables. A move to Factory el camió es desplaça a la posició de la fàbrica trobada abans a initialize. La variable arrived at Factory serà certa quan hagi arribat a la fàbrica i llavors passarà al estat arrived Factory. En aquest estat es carregarà el camió (s'actualitzarà la variable càrrega) i es decrementarà el stock de la fàbrica. Finalment s'inicialitzaran les variables de la botiga (posició i objecte) per tal de que el camió pugui anar a descarregar a la botiga. En aquest cas el procés és similar, quan el camió arriba a la botiga, cal augmentar el stock amb la càrrega del camió. Si no hi ha espai al magatzem de la botiga, el camió s'espera fins que pugui descarregar-ho tot i finalment tornarà a l'estat inicial on buscarà la posició de la fàbrica.

Finalment, abans de simular, només queda definir el "món" on es mouran els agents i la seva situació.

- A *Worlds*, creem el món "MonFactory" amb les opcions per defecte.
- A *Simulation Elements – Situation*, creeu el mapa o espai on es mourà el camió i la situació inicial de la botiga i la fàbrica. Podeu clicar l'opció *Torus* a map option (si es mou fora del límit del mapa surt per la banda oposada). Podeu veure un exemple de Situació en la figura següent:



Ja podeu Simular (New Simulation) i fent run observar el correcte funcionament dels diferents agents (podeu controlar la velocitat de simulació així com el que fa cada agent (acció i valors de les variables)).

### Exercicis:

1. Obriu el fitxer ex1.xml i entengueu el funcionament del sistema comentat en l'enunciat.
2. Es vol desenvolupar un sistema multi-agent de transport de productes similar al exercici 1 però entre diferents fàbriques. Tenim una fàbrica de bicicletes (producte final) que utilitza productes intermitjos que produeixen altres fàbriques nostres. Per simplicitat direm que la nostra bicicleta la fabriquem a partir de 2 productes intermitjos: el quadre i les rodes. Aquestes bicicletes cal vendre-les finalment a una botiga (també nostra). Cal dissenyar el sistema multi-agent en SeSAM perquè gestioni eficientment la fabricació i transport dels productes. En el sistema tindrem els següents agents:
  - Fàbrica quadres. Fàbrica que produeix 15 quadres a l'hora (hora= iteració) i una capacitat d'emmagatzematge de 100 productes. Si arriba a la capacitat màxima no pot produir més (no té on posar-ho).
  - Fàbrica rodes. Fàbrica que produeix 15 rodes a l'hora i una capacitat d'emmagatzematge de 100 productes. Si arriba a la capacitat màxima no pot produir més (no té on posar-ho).
  - Agent camió. Agent que transporta el producte X (roda, quadre o bicicleta) de la fàbrica de X a la fàbrica de bicicletes o botiga. Té una capacitat màxima de transport de 100 productes. Quan la fàbrica/botiga té espai lliure el camió li porta producte. Podeu tenir tants agents camió com vulgueu.
  - Fàbrica bicicletes. Agent/recurs que munta bicicletes. Té una velocitat de muntatge de 10 bicicletes x hora i una capacitat d'emmagatzematge de 100 productes intermitjos (quadres o rodes) i de 50 productes finals (bicicletes). Té un control del número de bicicletes produïdes. Si no té espai de producte final ha d'aturar la producció.
  - Botiga bicicletes. Agent/recurs que ven la bicicleta rebuda de fàbrica. Té un nivell de ventes de 15 productes x hora (hora=iteració) i una capacitat d'emmagatzematge de 100 productes. Té un control del número d'articles venuts.

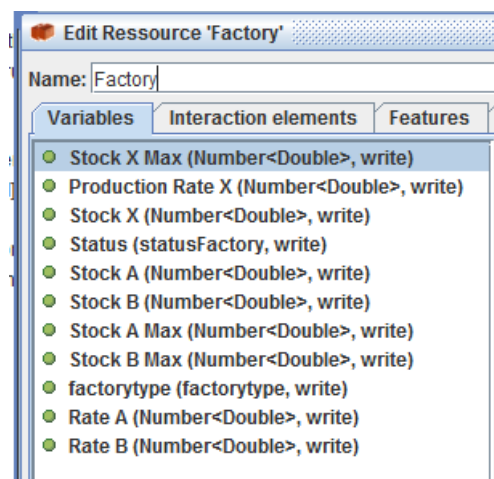
Nota: per aquest exercici vegeu solucions de PACs anteriors així com els tutorials sobre interacció entre agents, en concret:

- Com crear recursos (la fàbrica i botiga poden ser recursos).  
<http://130.243.124.21/mediawiki/index.php/CheeseClassResources>
- Com fer que un agent vagi cap a un recurs o interacció entre agents i recursos.  
<http://130.243.124.21/mediawiki/index.php/InteractionsBetweenResourcesAndAgents> i les funcions *GetFirst*, *ObserveObjectsOnPosition* i l'opció d'executar *Entry actions*, accions que cal fer abans d'executar una activitat).
- De la biblioteca d'exemples del SESAM (File- Open Model Library), vegeu l'exemple *cleanerworld*.

Hi han diferents opcions de disseny per a desenvolupar el sistema: Des de crear diferents agents camió per a diferents rutes i diferents agents fàbrica, fins a modificar els agents existents per fer-los prou genèrics per a desenvolupar les diferents tasques. En aquesta solució s'ha optat per definir un sistema prou genèric on es puguin posar

tantes fàbriques i camions com es vulgui. Per això s'han modificat els agents truck i factory, de la següent manera.

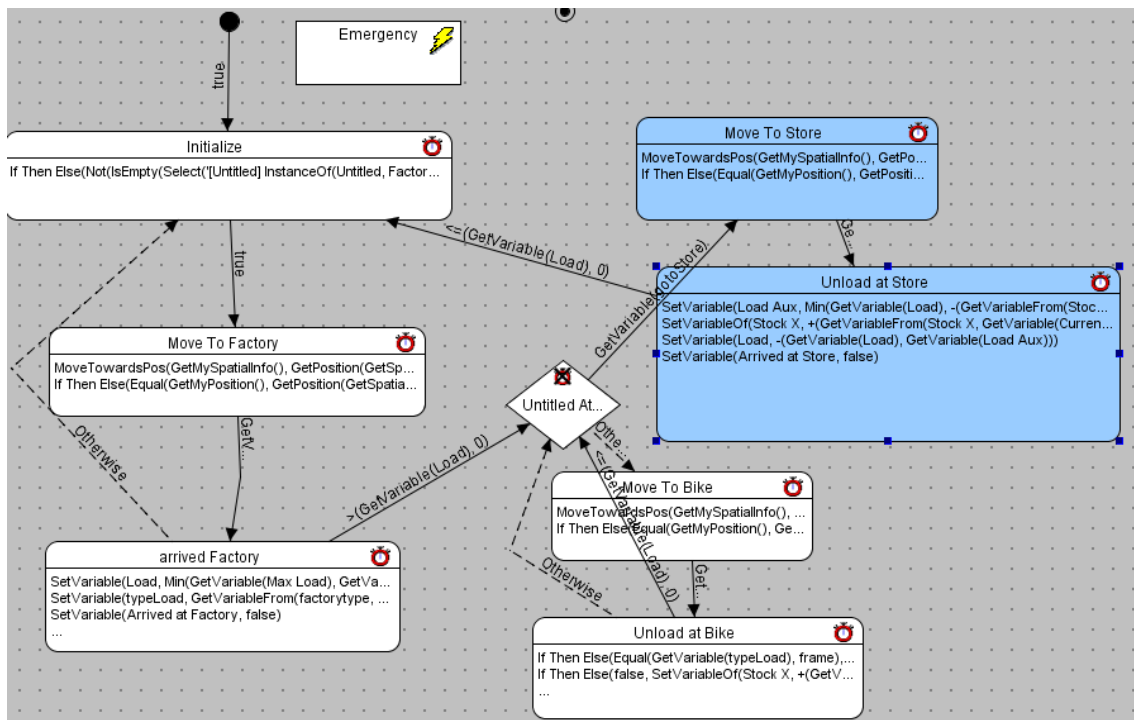
- **Factory:** Agent fàbrica genèric que pot produir bicicletes, rodes o quadres. Té un stock X (producte final de la fàbrica) però també stock A i B que és l'estock de productes intermitjos. En el cas de la fàbrica de bicicletes, A es refereix a quadres i B a rodes. La fàbrica produirà (canvia a estat Working) sempre que tingui stock suficient de A i B (donat per el consum necessari de cada producte A i B per fer X) i tingui suficient espai per el producte final X. El consum de A i B es representa per les variables Rate A i Rate B (1 quadre i 2 rodes en l'exemple per produir la bicicleta). L'estat es modificarà al Next Value de la variable *Status*. També tenim factorytype que ens diu el que produeix aquella fàbrica (serà utilitzat per l'agent truck per saber quina càrrega porta). La següent figura mostra les noves variables de l'agent factory. Els Stocks A, B i X es modificaran als Next Values pertinents de cada variable en el cas que la fàbrica estigui produint.



- **Truck:** L'agent camió anirà a una fàbrica (a l'atzar) a recollir el producte (Stock X) i el portarà a la fàbrica de bicicletes (si és frames o wheels) o a la botiga si el Stock X són bicicletes. Això es decideix a l'activitat *arrived Factory*. Un cop deixat el producte torna a anar a una altra fàbrica (si ha anat a la botiga) o a la botiga si estava a la fàbrica de bicicletes. Té noves variables on guarda l'objecte fàbrica de bicicletes i variables per controlar el fluxe de l'agent (gotoStore, anar a la botiga, o TypeLoad on es guarda el tipus de càrrega). En la següent figura es mostren les noves variables:

|   |           |
|---|-----------|
| Name: Truck   |           |
| Reasoning Engine  | Variables |
| <ul style="list-style-type: none"><li>Max Load (Number&lt;Double&gt;, const)</li><li>Position Of Factory (Position, write)</li><li>Load (Number&lt;Double&gt;, write)</li><li>Arrived at Factory (Boolean, write)</li><li>Currently Observed Factory (SimObject, write)</li><li>Arrived at Store (Boolean, write)</li><li>Currently Observed Store (SimObject, write)</li><li>Position Of Store (Position, write)</li><li>Load Aux (Number&lt;Double&gt;, write)</li><li>BikeFactoryObject (SimObject, write)</li><li>i (Number&lt;Integer&gt;, write)</li><li>gotoStore (Boolean, write)</li><li>typeLoad (factorytype, write)</li></ul> |           |

Les activitats de l'agent truck es mostren en la figura següent on s'observa el funcionament de l'agent comentat anteriorment.



3. Discutiu (no cal implementar-ho) com faríeu més eficient el sistema, quins agents hi posaríeu, justifiqueu la resposta.

El sistema plantejat és prou genèric per funcionar amb un número variable de camions i fàbriques, el problema d'això és que no és del tot eficient ja que l'assignació de fàbriques es realitza de manera aleatòria i pot passar que el camió que arribi a una fàbrica aquesta no tingui producte, que molts camions vagin a la mateixa fàbrica o que durant un temps no n'hi vagi cap. En aquest sentit, caldria dotar els agents truck de més coneixement (comprovar l'estat de totes les fàbriques i el seu stock) o tenir un agent coordinador que seria el que decidiria la fàbrica que hauria d'anar el camió. Un altre factor important de l'eficiència del sistema és el dimensionament correcte del magatzem de stocks així com el número de fàbriques de cada tipus (segurament caldrien més fàbriques de rodes que no de quadres).

**Cal entregar la memòria i el fitxer de model (xml) de l'exercici 2.**