PAC 3

Presentació

Tercera activitat d'avaluació continuada del curs. En aquesta PAC es pretén conèixer i desenvolupar sistemes multiagent.

Competències

Competències de grau

- Capacitat per utilitzar els fonaments matemàtics, estadístics i físics i comprendre els sistemes TIC.

- Capacitat per analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per abordar-lo i resoldre'l.

- Capacitat per conèixer les tecnologies de comunicacions actuals i emergents i saber-les aplicar, convenientment, per dissenyar i desenvolupar solucions basades en sistemes i tecnologies de la informació

- Capacitat per proposar i avaluar diferents alternatives tecnològiques i resoldre un problema concret

Competències específiques

- Capacitat per utilitzar la tecnologia d'aprenentatge automàtic més adequada per a un determinat problema.

- Capacitat per avaluar el rendiment dels diferents algorismes de resolució de problemes mitjançant tècniques de validació creuada.

Objectius

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 realitzar un nou sistema multi-agent.

Solució de la PAC

SeSAm. Tutorial exemple Ratolí

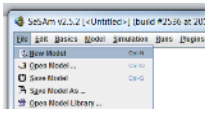
El tutorial s'ha extret de

http://130.243.124.21/mediawiki/index.php/TutorialIndex

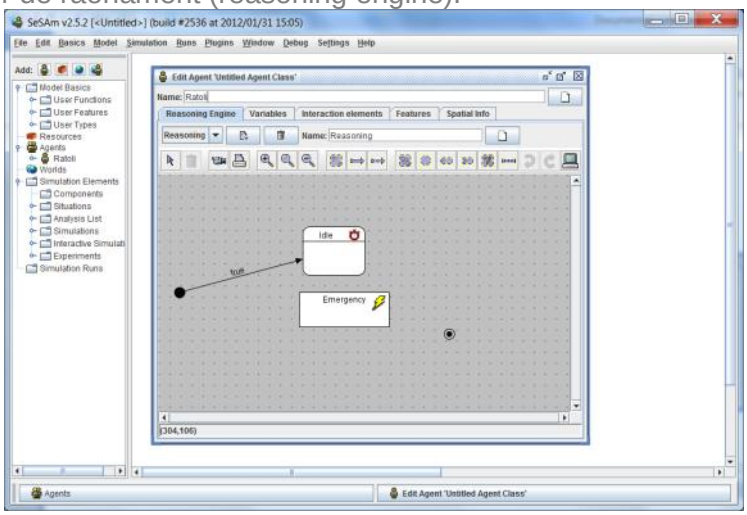
Volem simular el moviment d'un agent ratolí. Per simplificar-ho, el nostre ratolí es mou per l'entorn de manera que la velocitat del ratolí depèn del seu cansament. Quan el ratolí està cansat s'adorm i es recupera (disminueix el seu cansament) per tornar a moure's. El ratolí també té una propietat que és la seva edat.

A continuació veurem els diferents passos per a crear un entorn amb SeSam amb diversos agents ratolins.

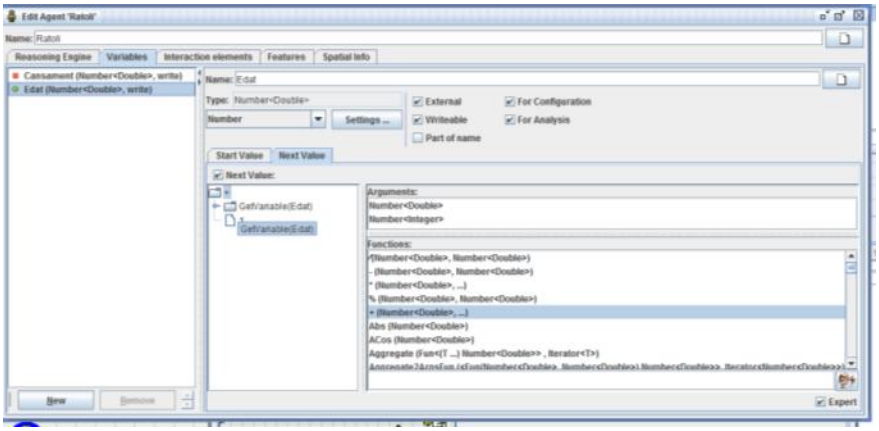
1. Obrir el SeSAm i crear un nou model buit. New model



2. **Crear un nou agent**: "Ratolí". Al clicar a dins l'agent s'obre la finestra per definir-ne les seves propietats, ens centrarem en les variables i el motor de raonament (reasoning engine).



- Variables. Crear les noves variables.
  - Cansament. Propietats Writeable (es pot escriure) amb el valor inicial (ex. 36) i externa.
  - Edat. Writable i externa amb valor inicial 0. A la pestanya *next value* cal especificar com varia el següent valor. Aquí i en els següents passos s'utilitzarà l'editor de funcions que permet especificar funcions de manera visual (però un mica feixuga). L'objectiu és sumar 1 a l'edat en cada iteració, per tant clicarem a la funció +, amb el primer operador la variable edat (GetVariable(Edat)) i l'altre operador 1.

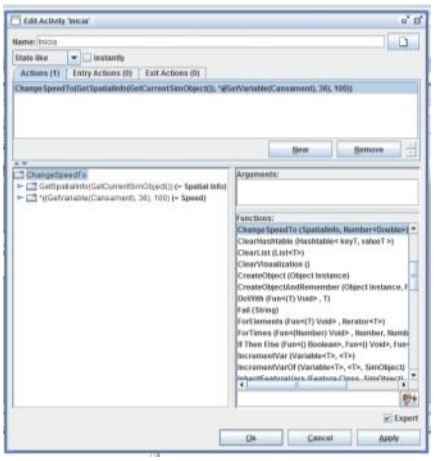


3. **Crear Activitat.** El ratolí podrà fer les següents activitats
   - Inicia. Inicialitzar la velocitat inicial depenent del seu cansament.
   - Mou. Moviment del ratolí (es cansa).
   - Dorm. Dormir (descansa)

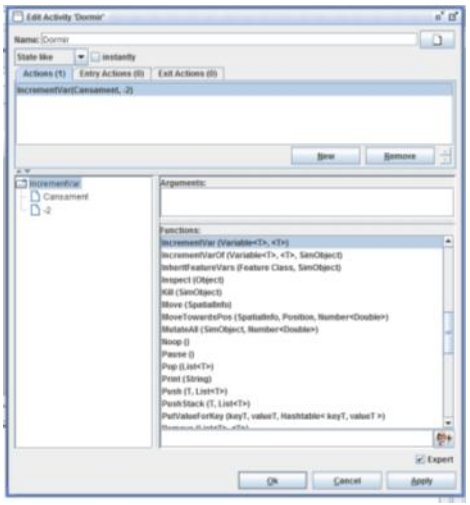
Les activitats s'especifiquen en l'apartat Reasoning Engine (motor de raonament). Bàsicament cada activitat es representa amb una caixa i conté una o més accions. Hi han dues accions "especials", la inicial (punt negre) i final (cercle amb un punt petit). L'agent s'inicia a partir de l'activitat inicial i depenent d'unes regles (fletxes) passarà a diferents activitats.

- **Activitat Inicia**. Editem l'activitat per defecte (idle). Especifiquem nova acció (apareix noop). Fem clic a expert mode i triem la funció ChangeSpeed. Hi definim l'objecte actual (GetSpacialInfo) i hi especificarem la velocitat amb la següent

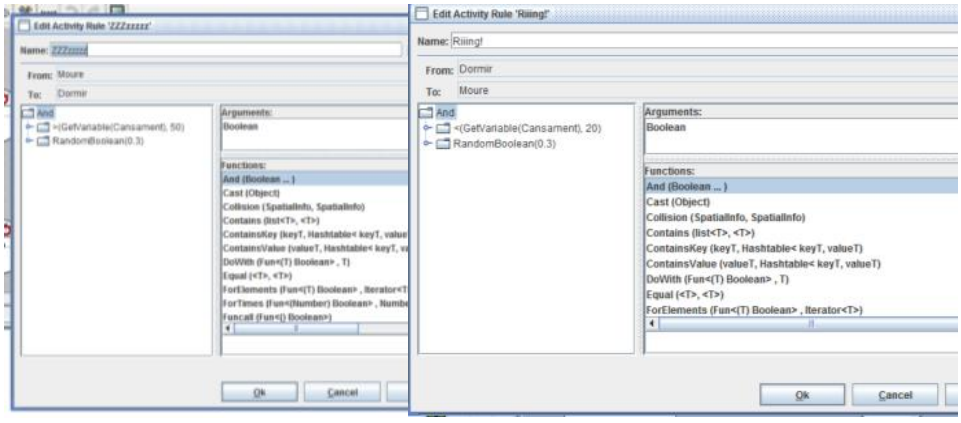
funció depenent del cansament, Cansament /36 * 100 amb la definició de funcions, tal i com mostra la figura: *(ChangeSpeedTo(GetSpatialInfo (GetCurrentSimObject()), *((GetVariable(cansament), 36), 100*))



- **Activitat Moure**. Mou el ratolí un angle i direcció determinat, i augmenta el cansament del ratolí. Creem una nova activitat (caixa rodona) on hi afegim 3 accions:
    - Canviar l'angle del moviment: ChangeDirectionBy(GetSpatialInfo(GetCurrentSimObject()), If Then Else(RandomBoolean(0.5), /(GetVariable(cansament), 9), -Unary(/(GetVariable(cansament), 9))))
    - Moure: Move(GetSpatialInfo(GetCurrentSimObject()))
    - Canviar la variable cansament: SetVariable(cansament, + (GetVariable (cansament), 1))
- **Activitat Dormir.** Quan el ratolí dorm redueix el seu cansament. En aquest cas creem una acció que modificarà la variable cansament.

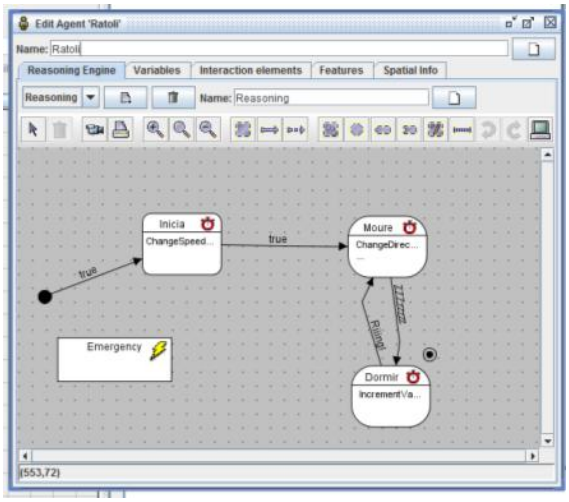


4. **Crear Regles.** Condicions perquè el ratolí passi d'una activitat a l'altra. Es creen amb les fletxes.
    - De Inicia a Moure es compleix sempre (per tant serà sempre a true).
    - De Moure a Dormir (ZZZZzzzz), quan estigui cansat (cansament >50).
    - De Dormir a Moure (Riiing!), quan ja no estigui cansat (cansament <20).



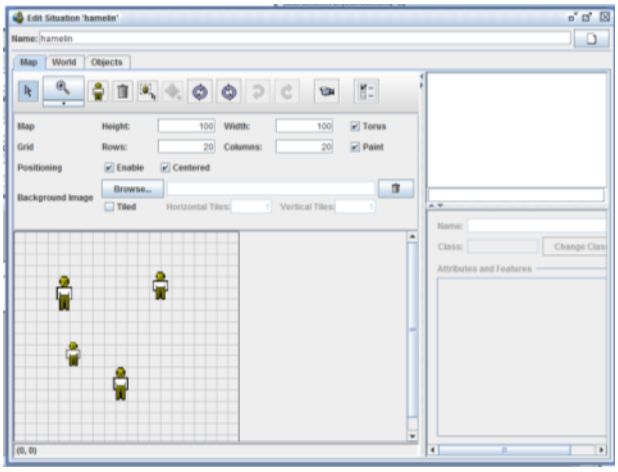
Regla ZZZZzzzz Regla Riiing!

Al final de tots aquests passos haurem construït el nostre agent ratolí, la finestra de definició de l'agent quedaria de manera similar a:



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

- A *Worlds*, creem el món "MonRatolí" amb les opcions per defecte.
- A *Simulation Elements – Situation*, creeu el mapa o espai on es mouran els ratolins i la situació inicial i número de ratolins que hi hauran. Assegureu-vos de 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,



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. Desenvolupar el model ratolí de l'exemple, simular-lo i veure el seu correcte funcionament.
2. Cal desenvolupar un sistema multi-agent de transport de productes d'una fàbrica a una botiga determinada amb les següents característiques.

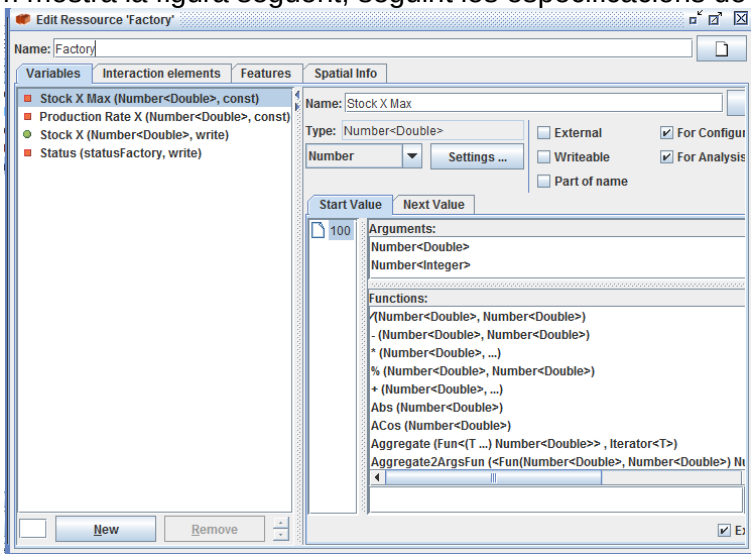
    - Agent camió. Agent que transporta el producte X de la fàbrica a la botiga. Té una capacitat màxima de transport de 25 productes. Quan la botiga té espai li porta producte.
    - Botiga. Agent/recurs que ven el producte X rebut de fàbrica. Té un nivell de ventes de 10 productes x hora (hora=iteració) 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).

    Nota: per aquest exercici vegeu el tutorial sobre interacció entre agents, en concret:

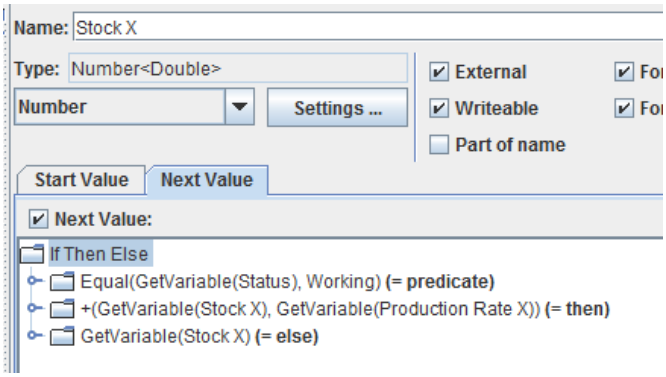
    - Com crear recursos (la fàbrica i botiga poden ser recursos). http://130.243.124.21/mediawiki/index.php/CheeseClassResources
    - Com fer que un agent vagi cap a un recurs o interacció entre agents i recursos. http://130.243.124.21/mediawiki/index.php/InteractionsBetweenResourcesAndAgents i les funcions *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*, el podeu utilitzar com a base.

Seguint els exemples i el tutorial esmentats, es crearan els següents Agents i Recursos:

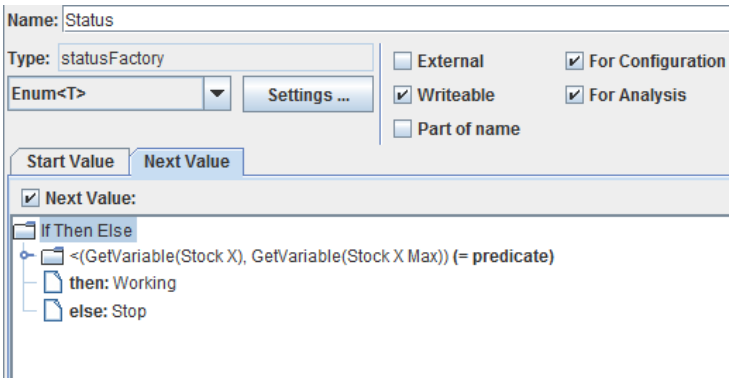
- **Fàbrica (Factory)**. Agent/recurs que tindrà les variables de stock, stock màxim, producció, estat en que es troba (fabricant, parat) tal i com mostra la figura següent, seguint les especificacions de l'enunciat.



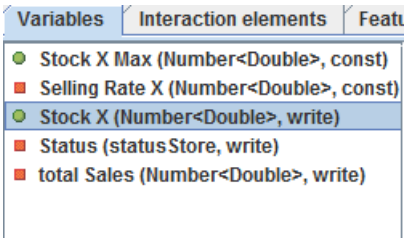
La variable stock X es va augmentant a cada iteració fins a arribar al stock màxim utilitzant l'opció next value d'aquesta variable, tal i com mostra la figura.



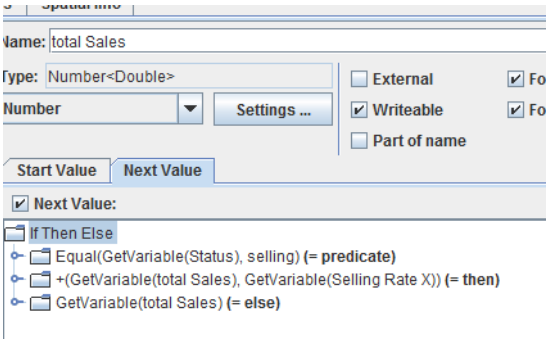
La variable d'estat (status) canviarà a produint o aturat depenent si s'arriba al stock màxim. Un altre cop s'utilitza l'opció next value, segons la figura següent.



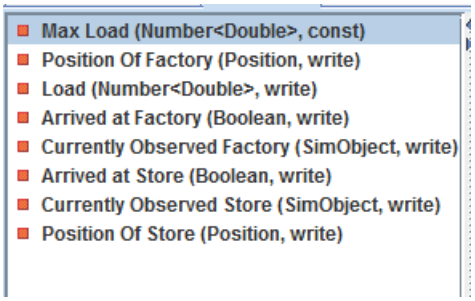
- **Botiga (Store)**. De manera similar a la fàbrica, conté les variables stock de producte, stock màxim, vendes/hora, estat (venent o parat). El stock es decrementarà a cada iteració (utilitzant next value) semblant al stock de la fàbrica però disminuint en comptes d'augmentar. De manera similar també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



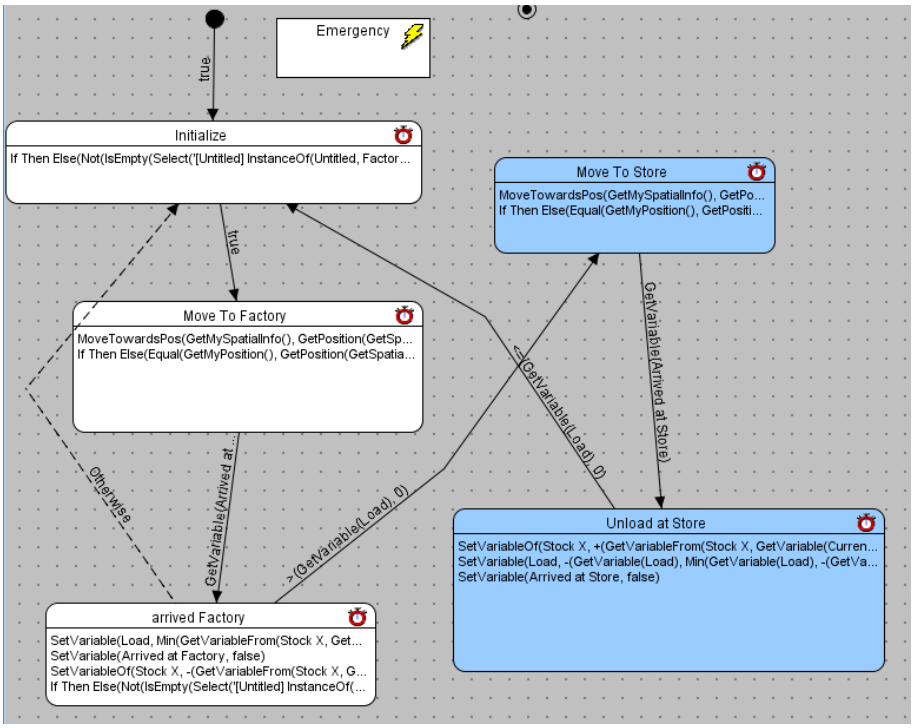
Hi ha una nova variable, vendes totals (total Sales) que augmentarà a cada iteració (amb el valor *selling rate*) si l'estat de la botiga és venent, tal i com mostra la figura:



- **Camió** (Truck). Agent que funcionarà de manera semblant però més senzilla que l'agent *cleaner* a l'exemple *cleanerworld.* Concretament, començant per una càrrega buida, anirà a la fàbrica a buscar el producte, quan hagi carregat (si hi ha prou stock a fàbrica) anirà a portar-lo a la botiga. El deixarà allà (si no té espai la botiga s'esperarà allà fins que pugui deixar tota la càrrega i tornarà a la fàbrica. Per a poder saber tota la informació de les fàbriques i les botigues l'agent tindrà les variables següents que mostra la figura:



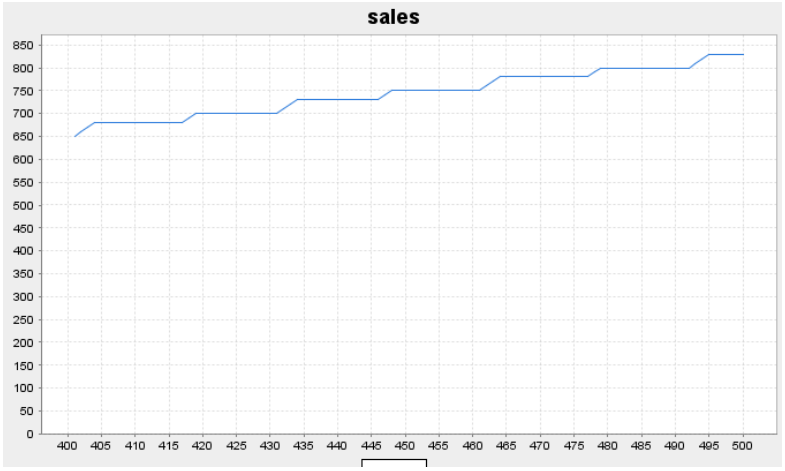
Cal destacar la càrrega màxima del camió (*Max Load*) i la càrrega actual (*Load*) així com les posicions, objectes referents a la botiga/fàbrica. S'ha afegit un booleà per confirmar quan s'hi ha arribat.
Tot el procés descrit anteriorment es troba desenvolupat en diferents *actions* de l'agent, tal i com mostra la figura.



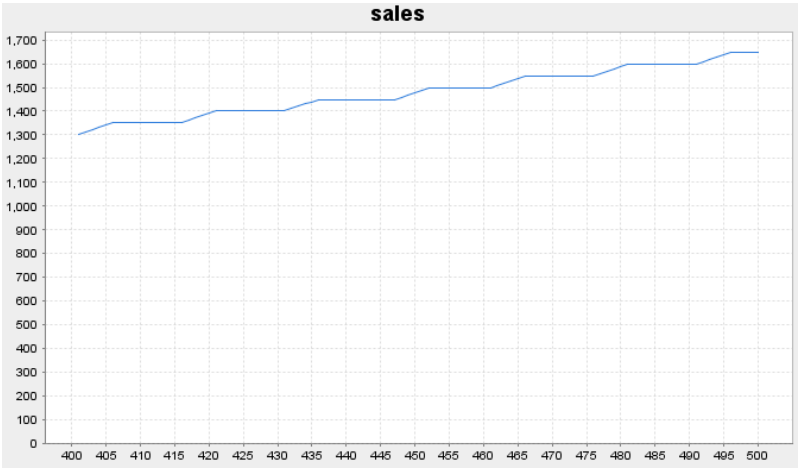
Les accions per anar a la botiga o a la fàbrica són similars bàsicament canviant els noms de les variables. A *move to Factory* el camió es desplaça a la posició de la fàbrica trobada abans a *initialize.* La variable *arrived at Factory* serà certa quan hagi arribat a la fàbrica i llavors passarà al 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.

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

Veiem que segons els paràmetres del sistema, la limitació és el transport, per tant, per poder augmentar les vendes caldria tenir més camions. Això seria fàcil de simular posant més agents camions en la simulació. La figura següent mostra les ventes totals amb 1 camió o amb dos camions després de 500 iteracions on s'aprecia la diferència (de 800 vendes comparat amb 1600 aproximadament).



Vendes amb 1 Camió



Vendes amb 2 Camions

S'adjunten els fitxers de model (xml) dels exercicis 1 i 2.