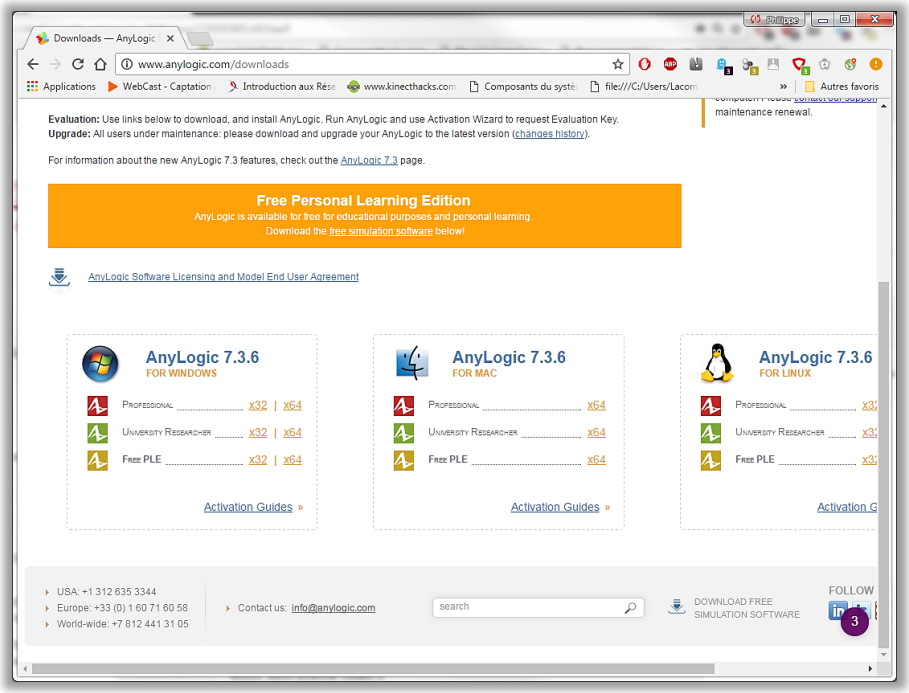
# Utilisation d'AnyLogic

**Auteurs :** P. Lacomme (placomme@isima.fr)
D. Lamy (lamy@isima.fr)
**Date de création :** Janvier 2017

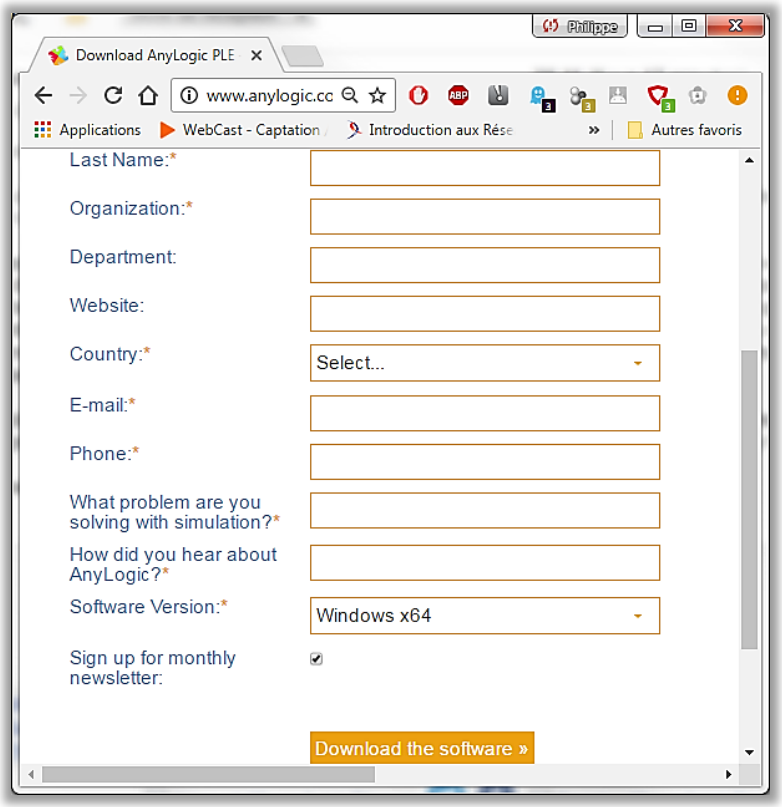
## 1) Installation

Le logiciel de simulation est disponible à l'adresse suivante : http://www.anylogic.com/
Le logiciel se trouve dans **Download** et il faut choisir Free PLE (Free Personal Learning Edition).


Il faut choisir la version correspondante au système utilisé (ici Windows 64 bits)



Il ne reste plus qu'à remplir le formulaire

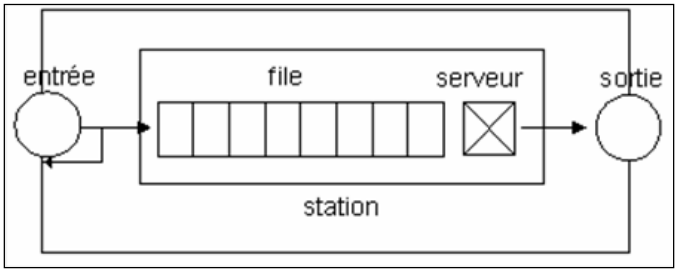


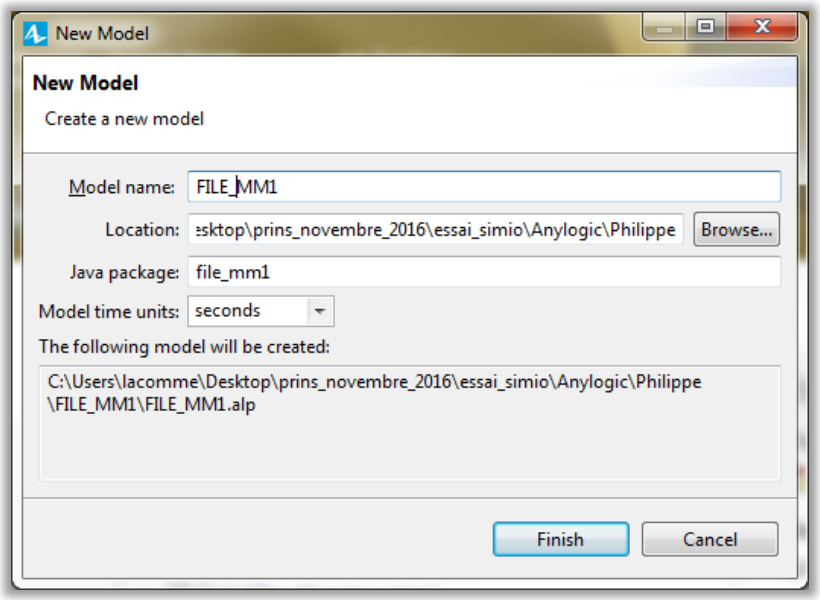

**Remarque :**

Contrairement à beaucoup d'autres environnements, Anylogic permet de réaliser plusieurs types de modèle de simulation. Il s'agit là du point fort de l'environnement.

**2) Modèle de simulation à évènements discrets : simuler une file MM1**

Le système à simuler se compose d'une source (entrée), d'une station (serveur + file d'attente) et d'une sortie.

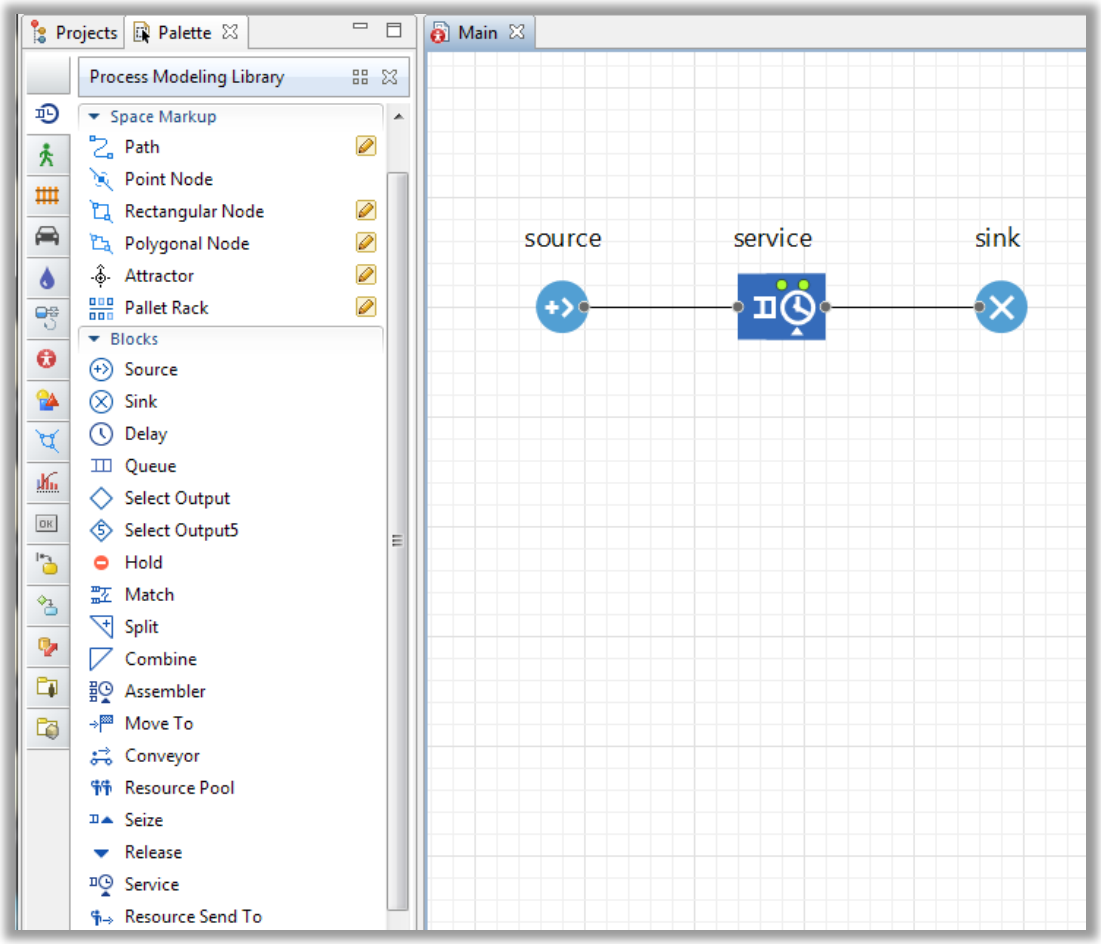




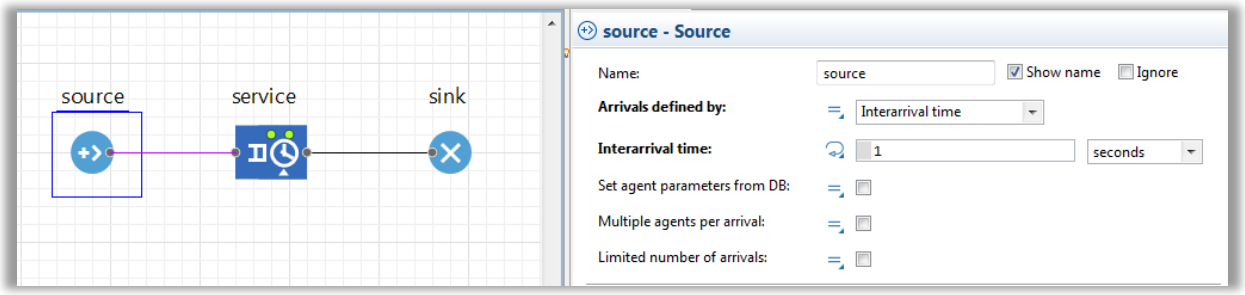

Trois éléments doivent être utilisés :

- une source ;
- un service ;
- un puits.

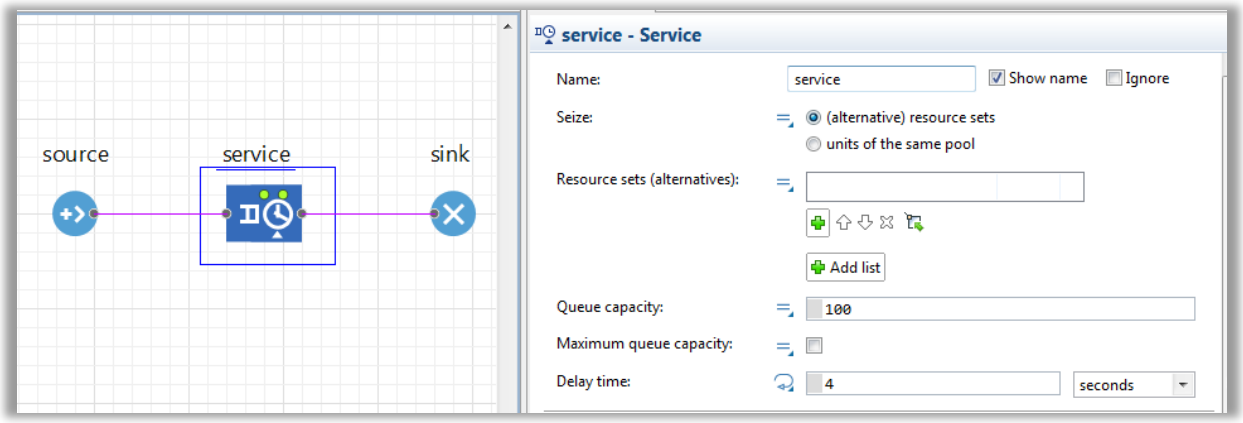
Ces éléments se trouvent dans **Process Modeling Library**.



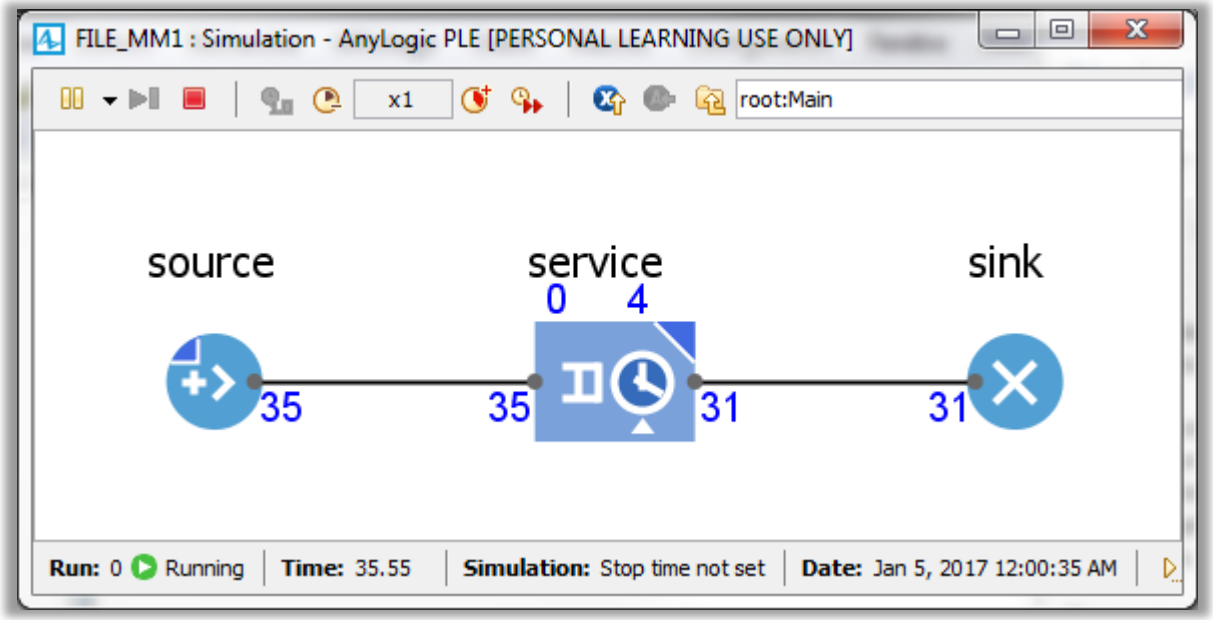
Pour la source, il faut spécifier que les arrivées sont définies par une durée interarrivée et que la durée interarrivée vaut 1.



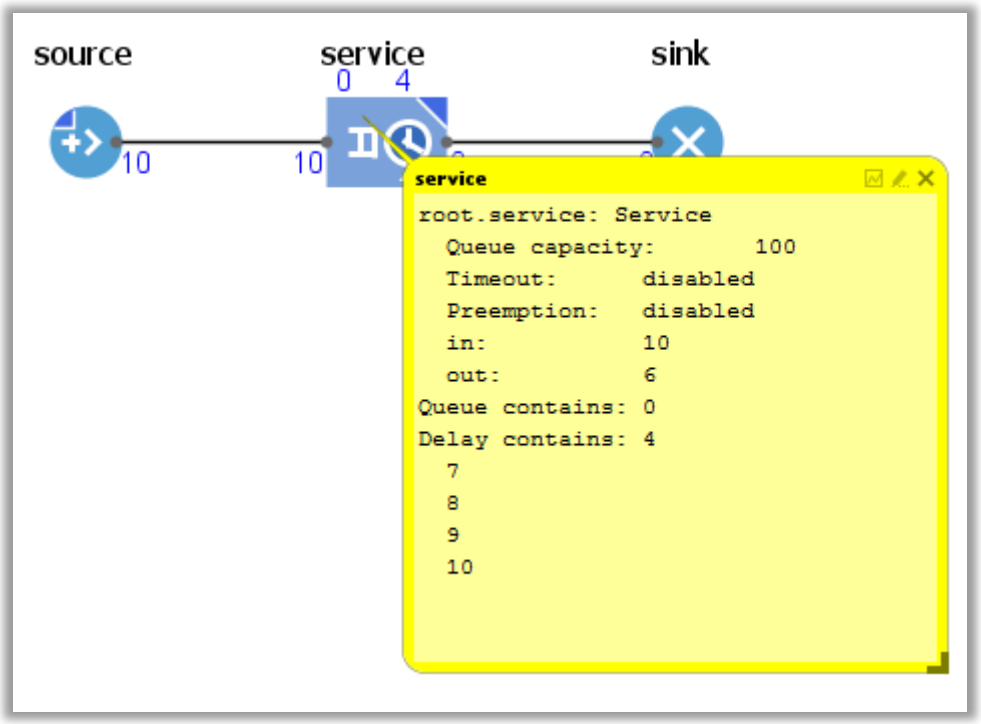

Au niveau du service, on peut spécifier une longueur de file t'attente (par exemple 100) et un temps de traitement (Delay Time) de valeur 4.



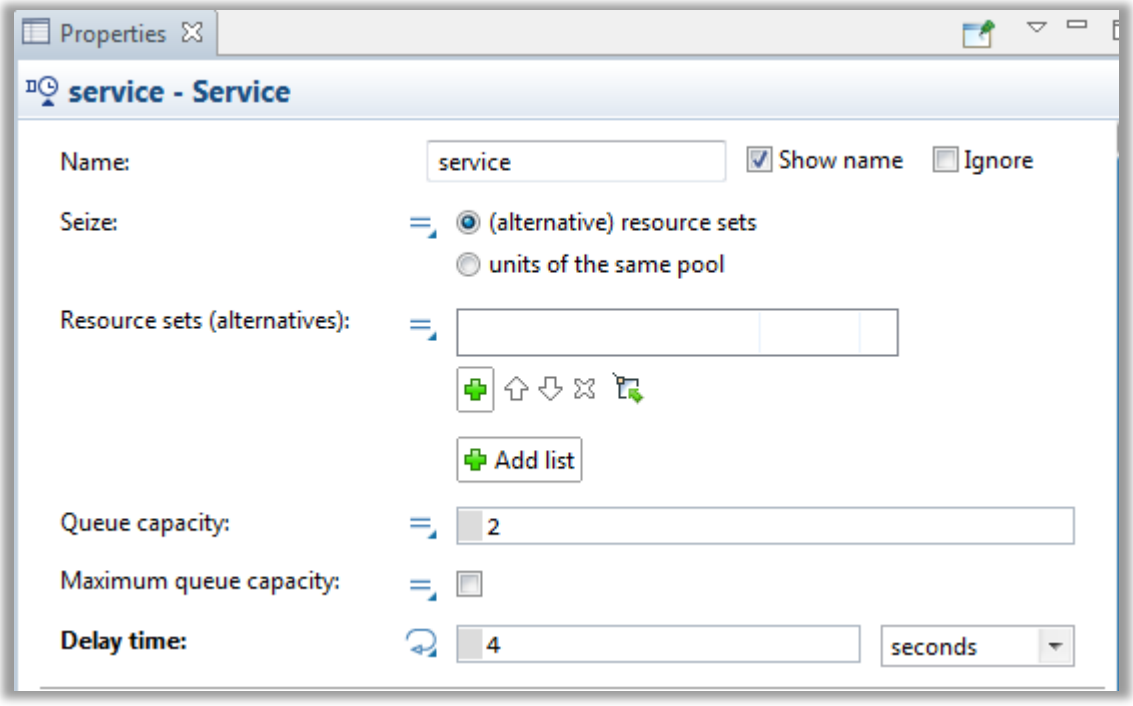

Le résultat d'exécution est le suivant :



On peut avoir accès aux informations sur le service (on dit couramment station dans la terminologie QNAP2) par un simple clic sur le service.




On peut limiter la taille de la file à 2, en modifiant les attributs du service.

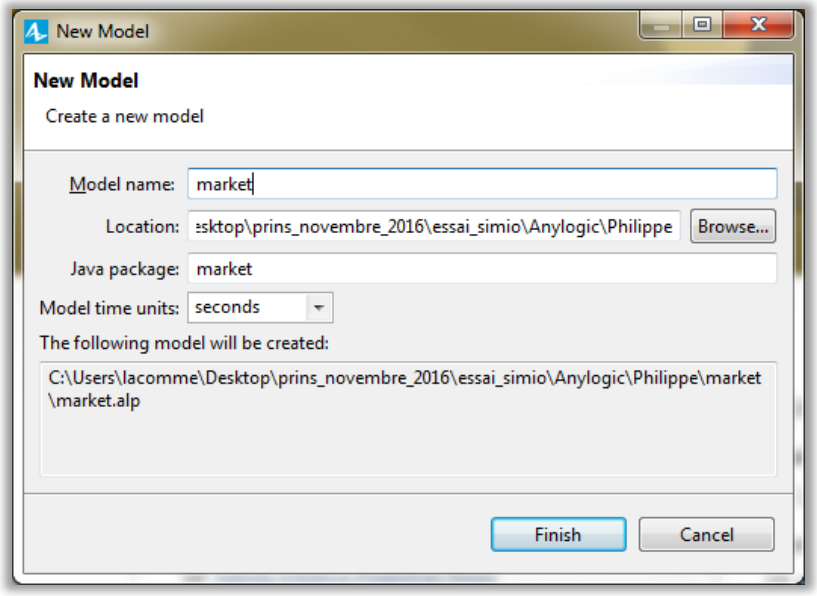



**3) Modèle de simulation « spatial »**

(Ce modèle est décrit dans le document officiel **anylogic-7-in-3-days.pdf**)

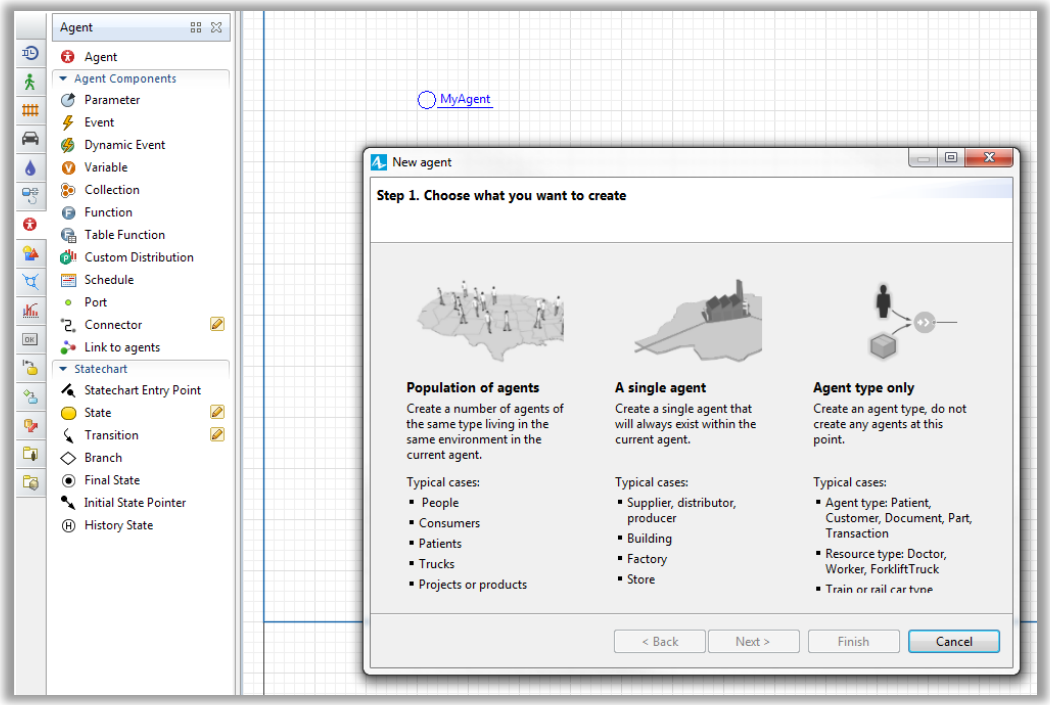
### 3.1. Création d'un modèle.

Il faut créer un projet nommé Market.

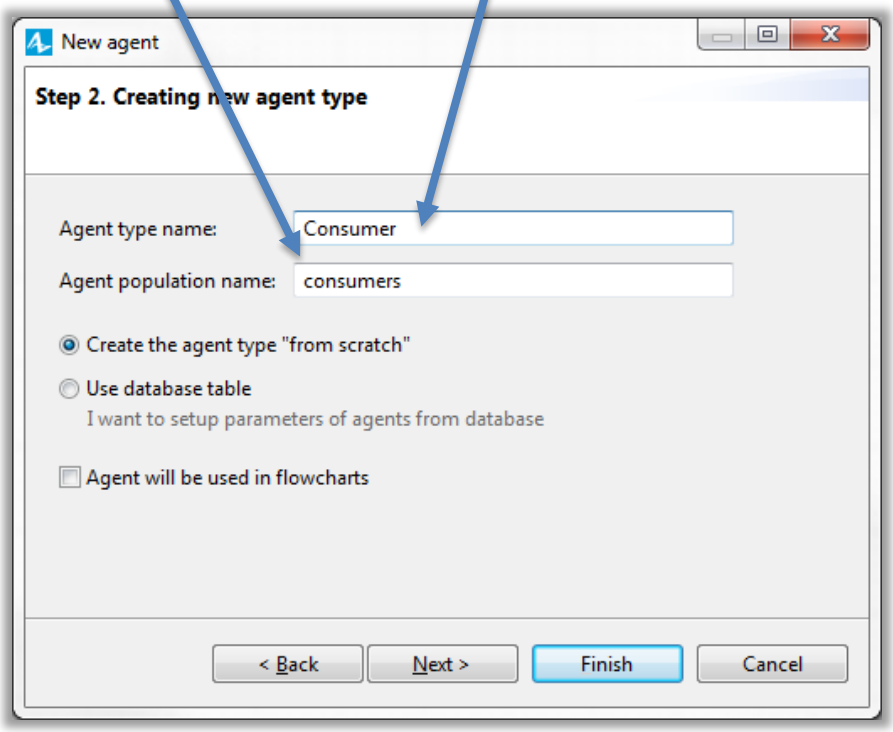



Ce projet va utiliser la notion d'agent. Un agent dans **Anylogic** peut correspondre à un élément de flux « classique » tel que nous l'avons dans un modèle de simulation à événements discrets.

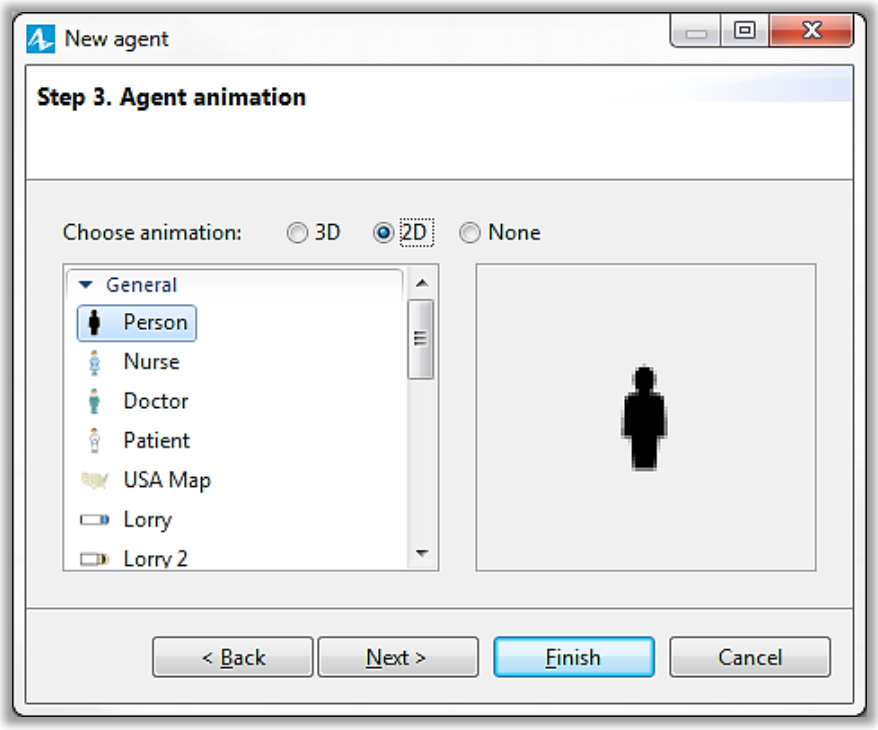
Il faut créer un Agent et le « déposer » sur la feuille. Il faut choisir Population of agents.



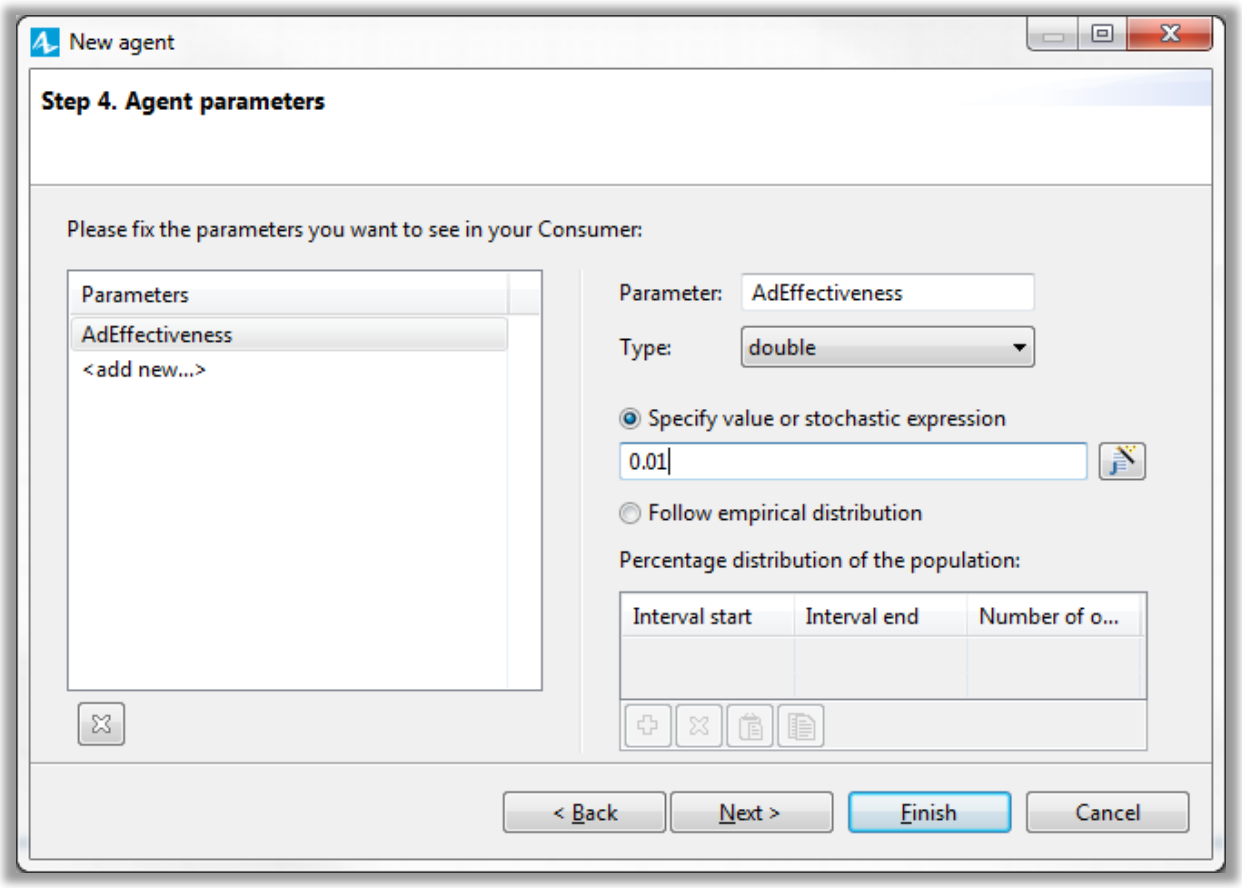
On peut ensuite choisir le nom de la classe Java ici **Consumers** et l'environnement AnyLogic complète avec le nom **consumers** pour la population.



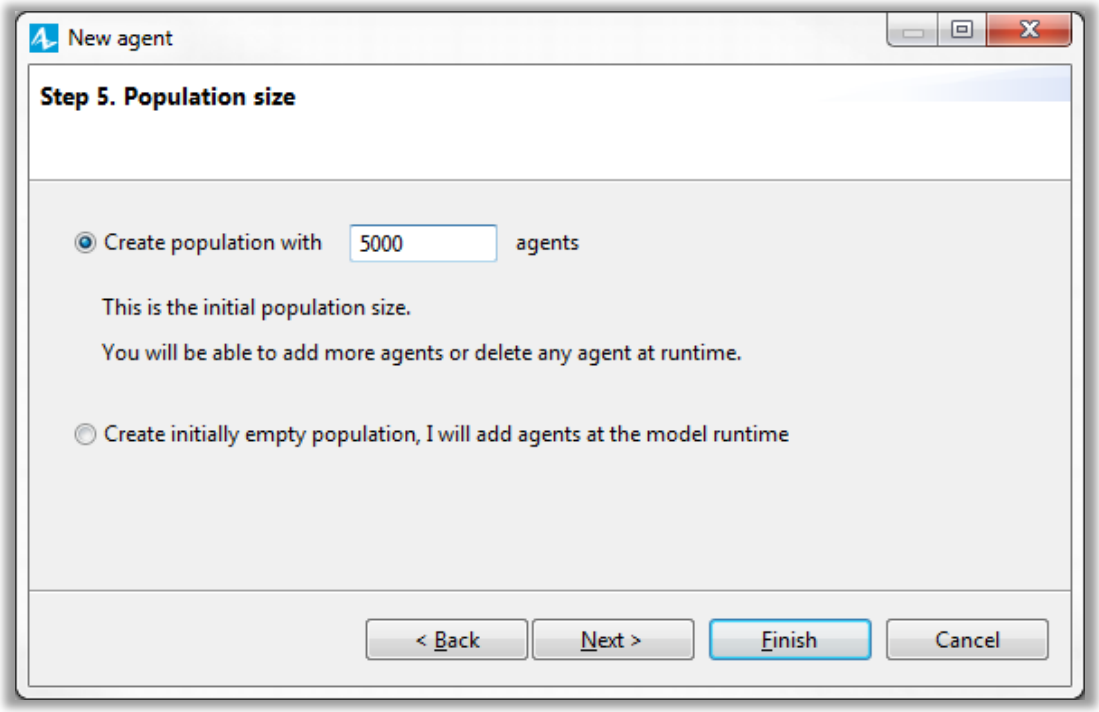

Ces deux notions sont très différentes comme nous le verrons par la suite.
On peut choisir ensuite sa représentation graphique en 2D.



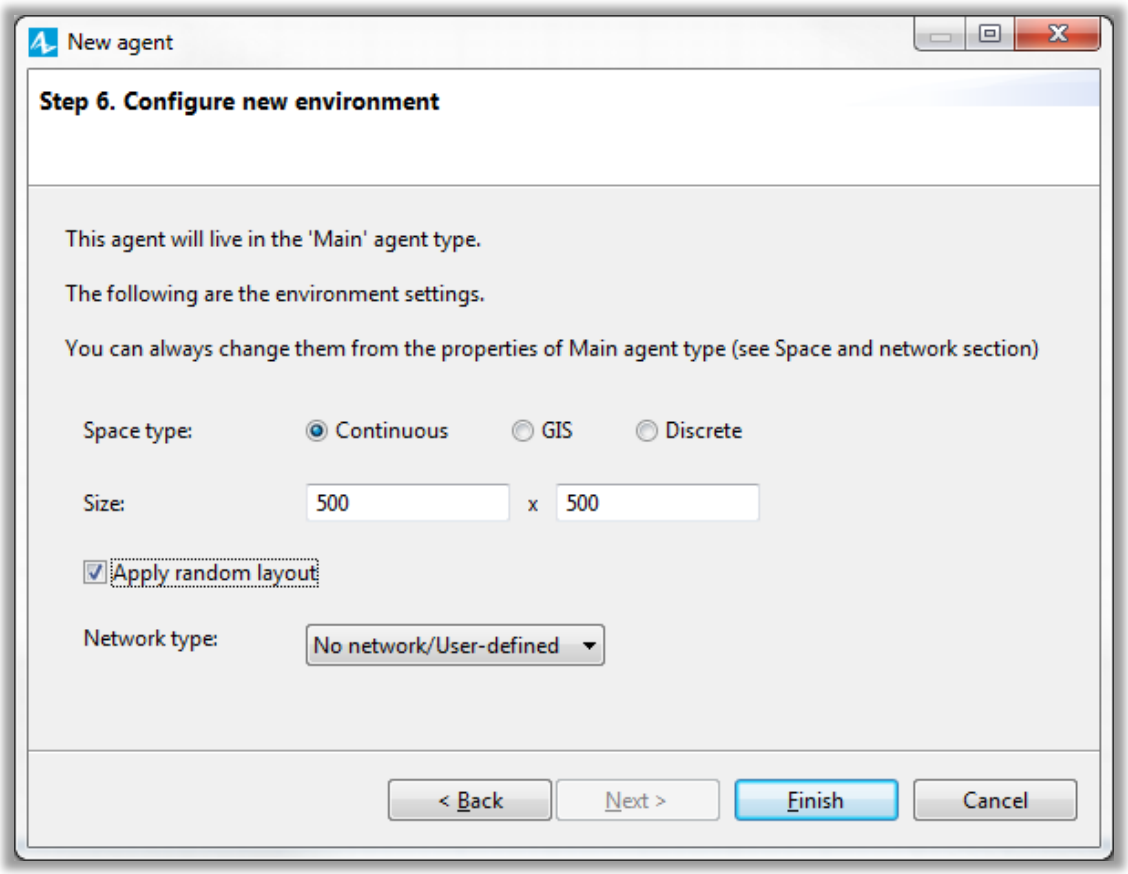

On va ajouter à la classe Consumer un attribut nommé AdEffectiveness avec une valeur par défaut de 0.01.



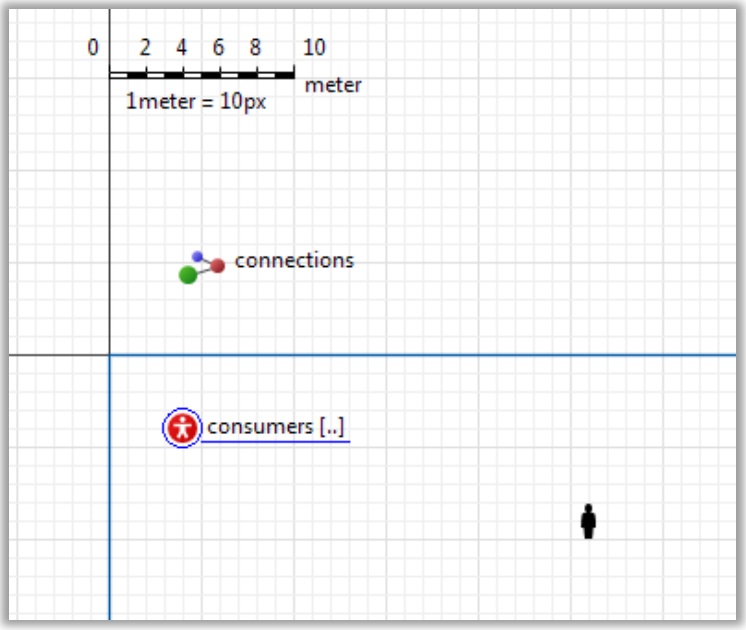

La population est alors constituée de 5000 agents.

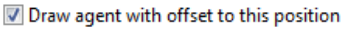


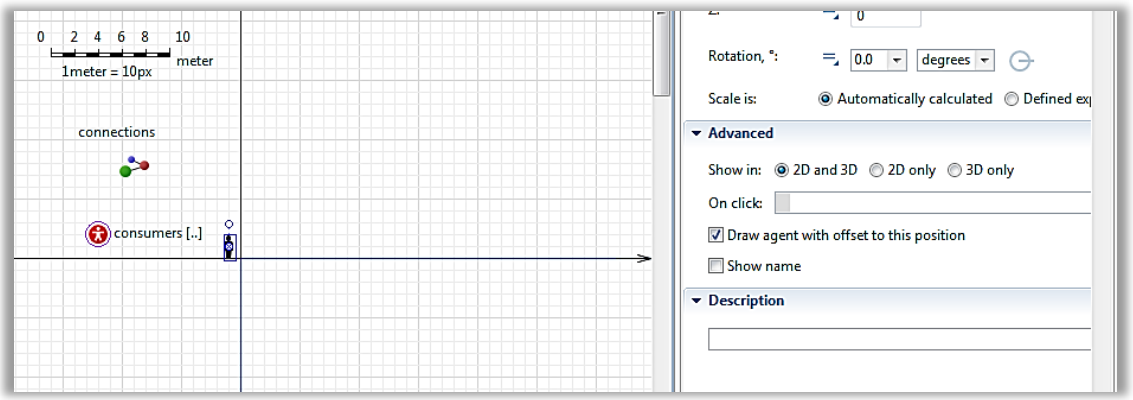

Comme on souhaite obtenir une représentation graphique, il faut définir un rectangle de par exemple 500 par 500.




Le modèle se présente alors comme suit :

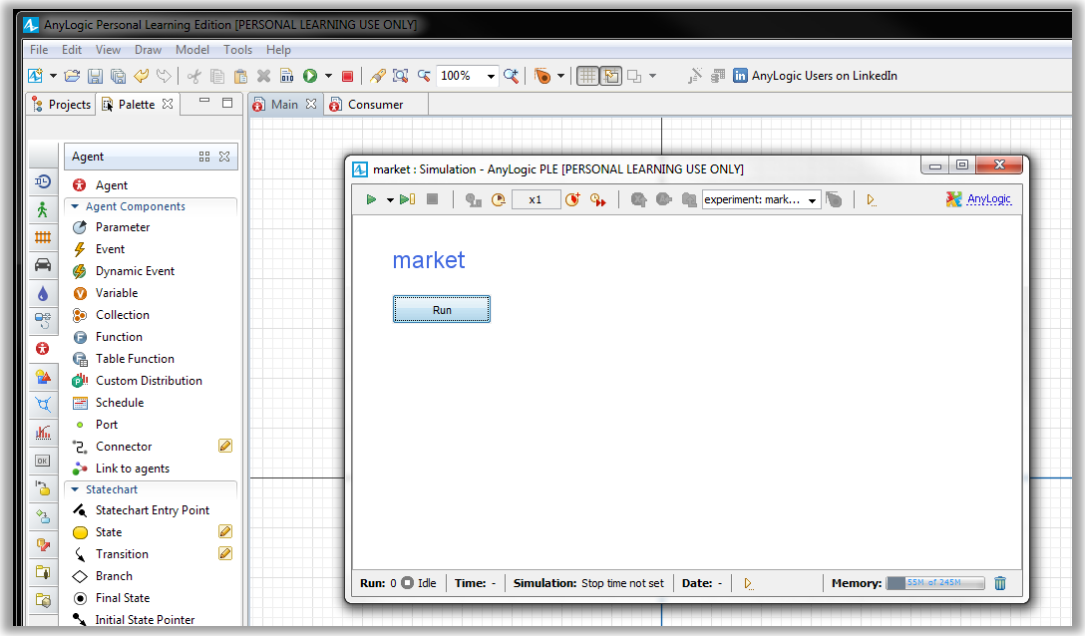



Il faut sélectionner l'icône et cocher la case 

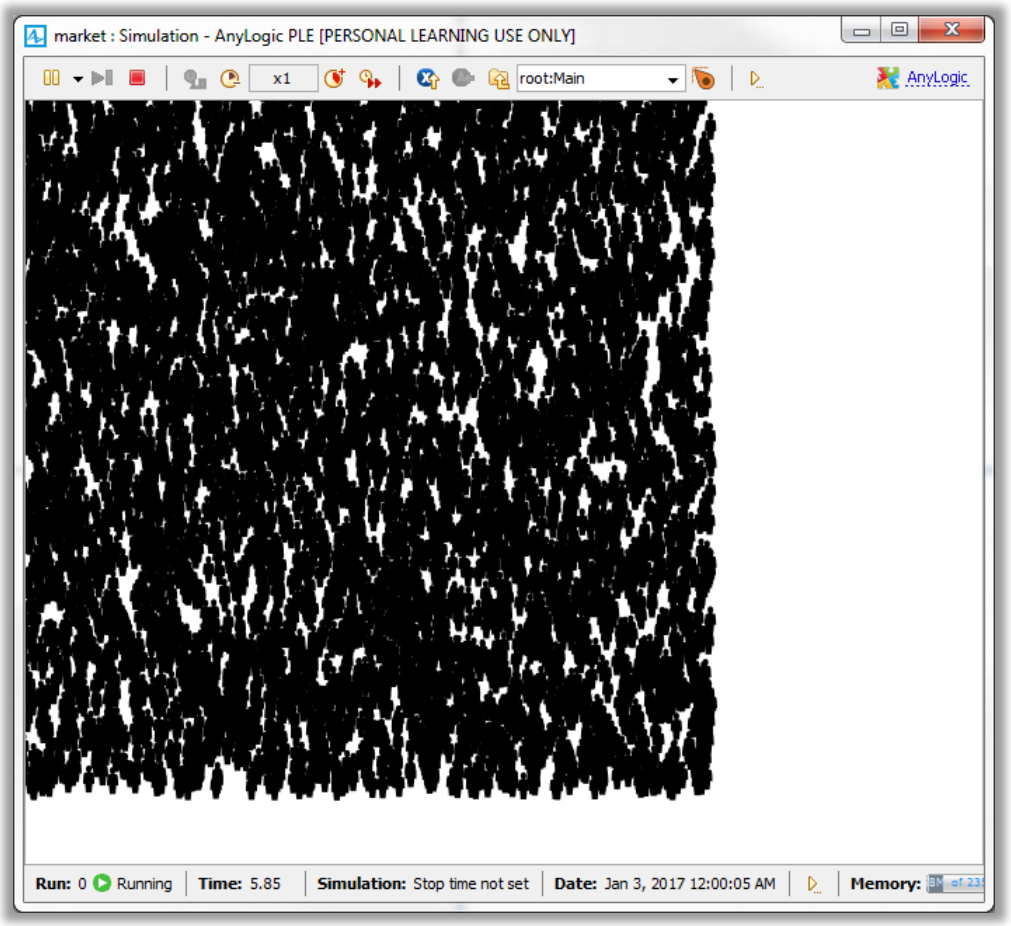



On peut ensuite vérifier le bon fonctionnement du modèle… en cliquant sur 

L'environnement génère le code java et lance l'application qui se présente comme ceci :

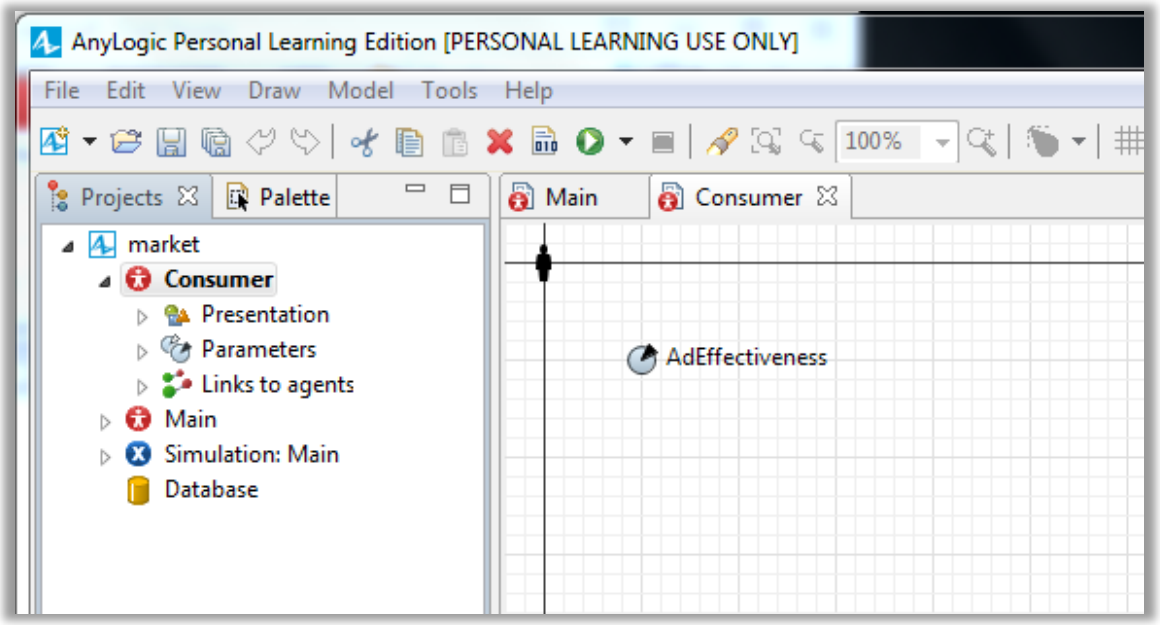



En utilisant le bouton **Run**, on obtient une première exécution.

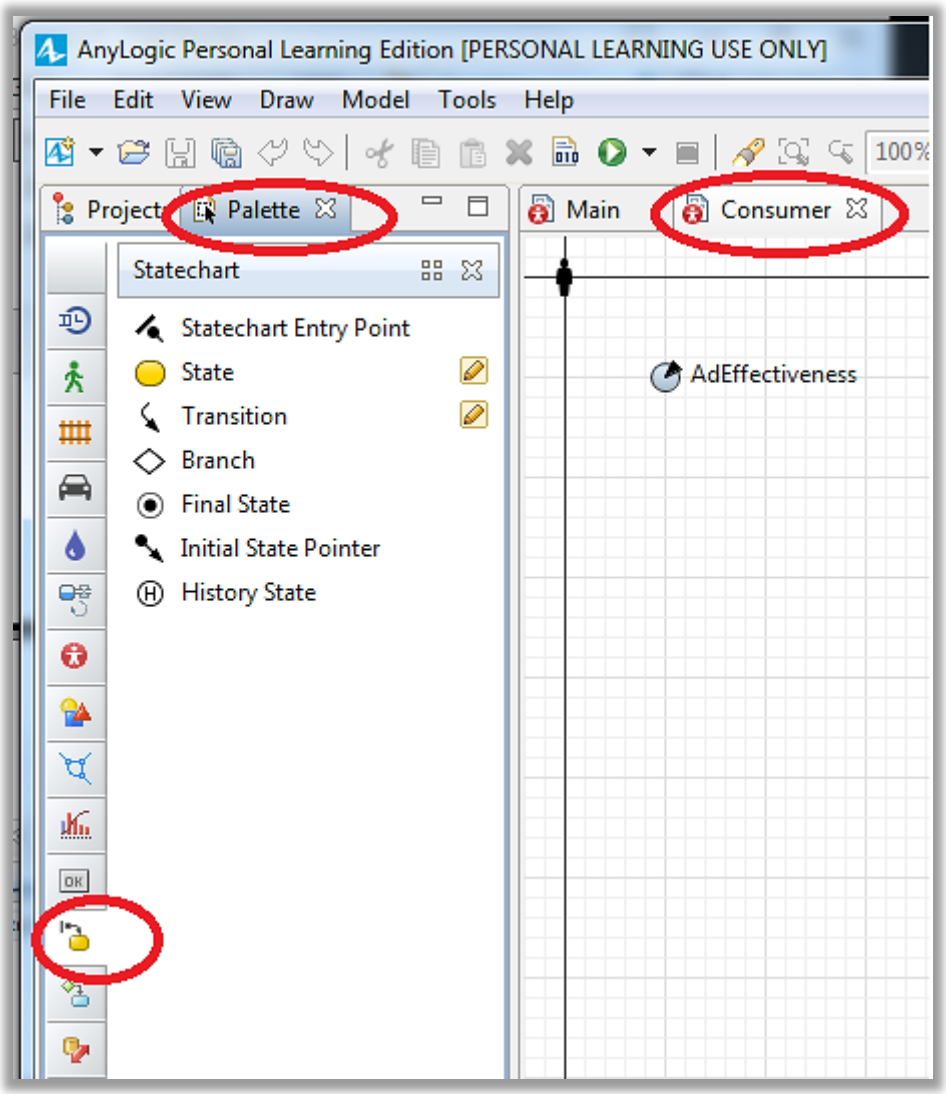



### 3.2. Définir un comportement du « consommateur »

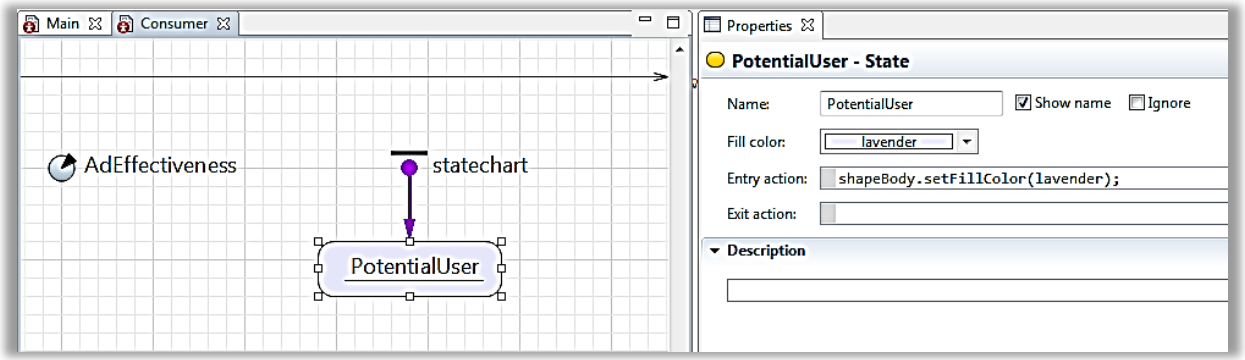
Il faut se placer dans l'onglet Consumer qui en réalité modélise la classe Consumer.



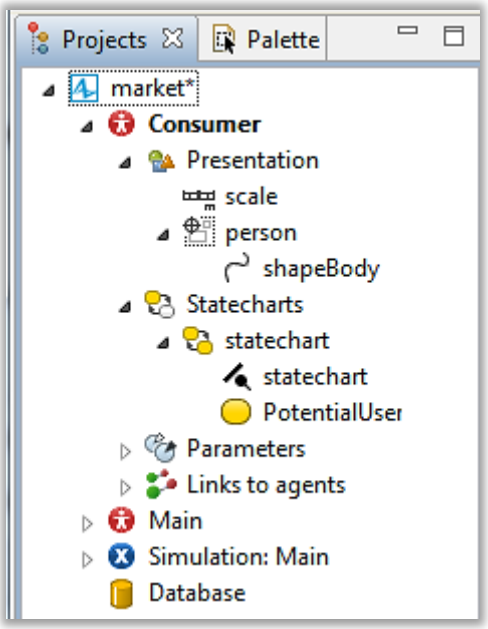
Définir le comportement du consommateur revient à définir un diagramme définissant les changements d'état : un **statechart**.
L'ensemble des objets nécessaires à la définition d'un diagramme d'état sont dans l'onglet :



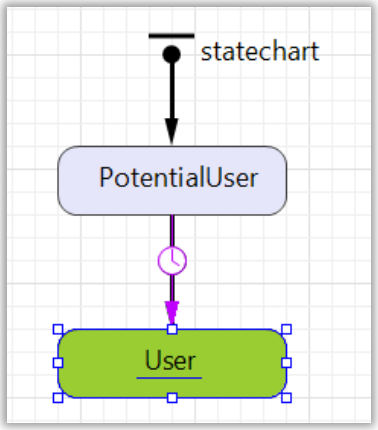

Dans un premier temps, il faut définir le point d'entrée et un premier état (nommé **PotentialUser**) avec une couleur par défaut (lavender).

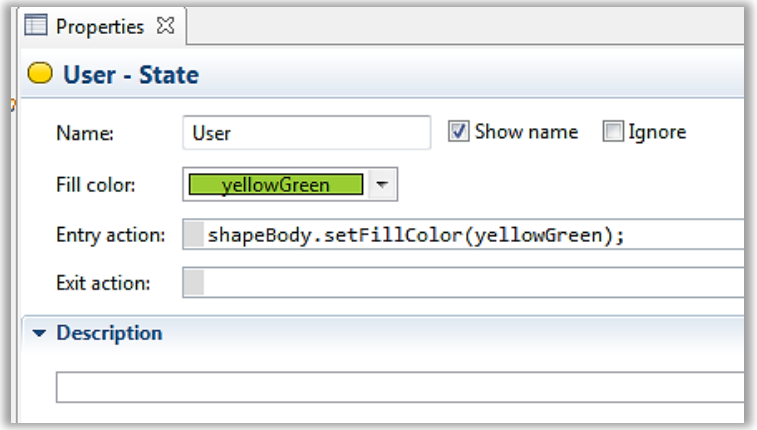


On retrouve l'ensemble des éléments ajoutés dans le projet.

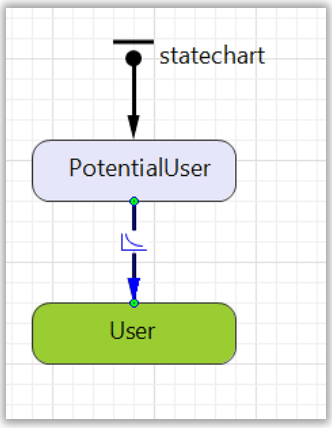


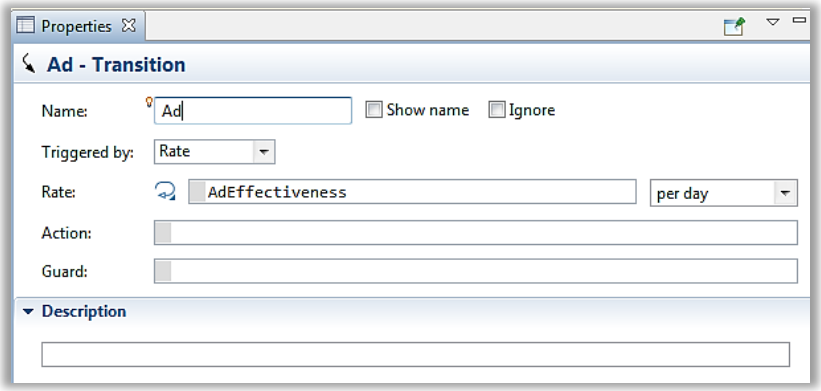

On peut ajouter un deuxième état "user" auquel on peut associer une couleur verte.



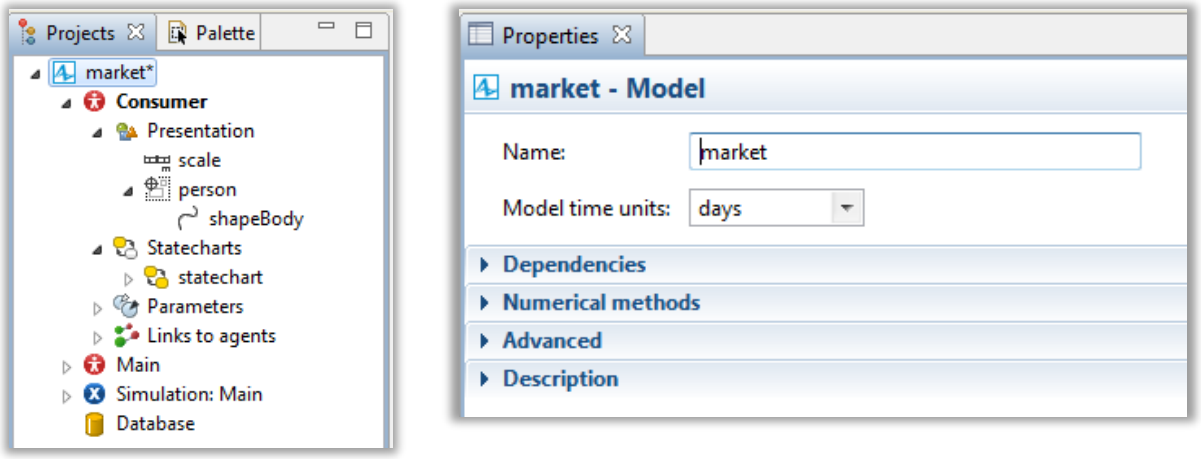




On sélectionne la relation entre PotentialUser et User pour définir les conditions de mise en œuvre.
Le trigger qui déclenche la mise à feu de la transition est donné par **AdEffectiveness**.

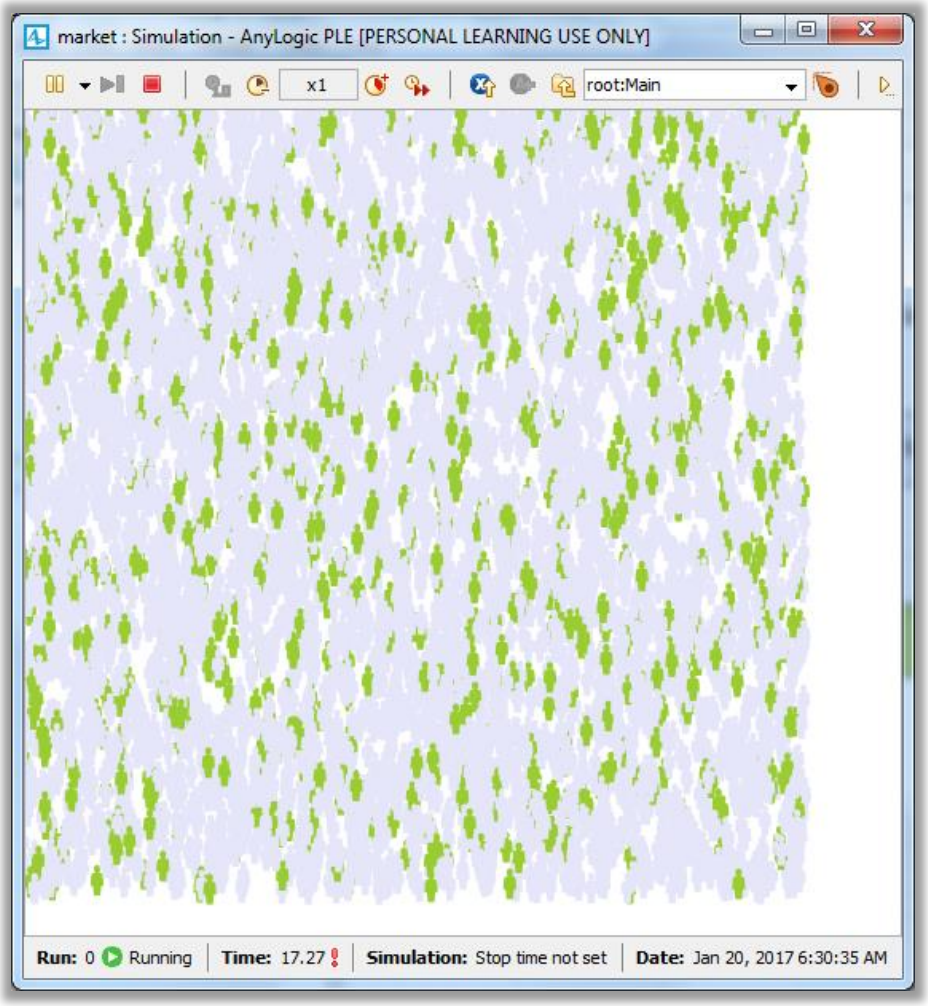


Il faut ensuite modifier le modèle pour choisir l'unité avec laquelle le modèle s'exécute.



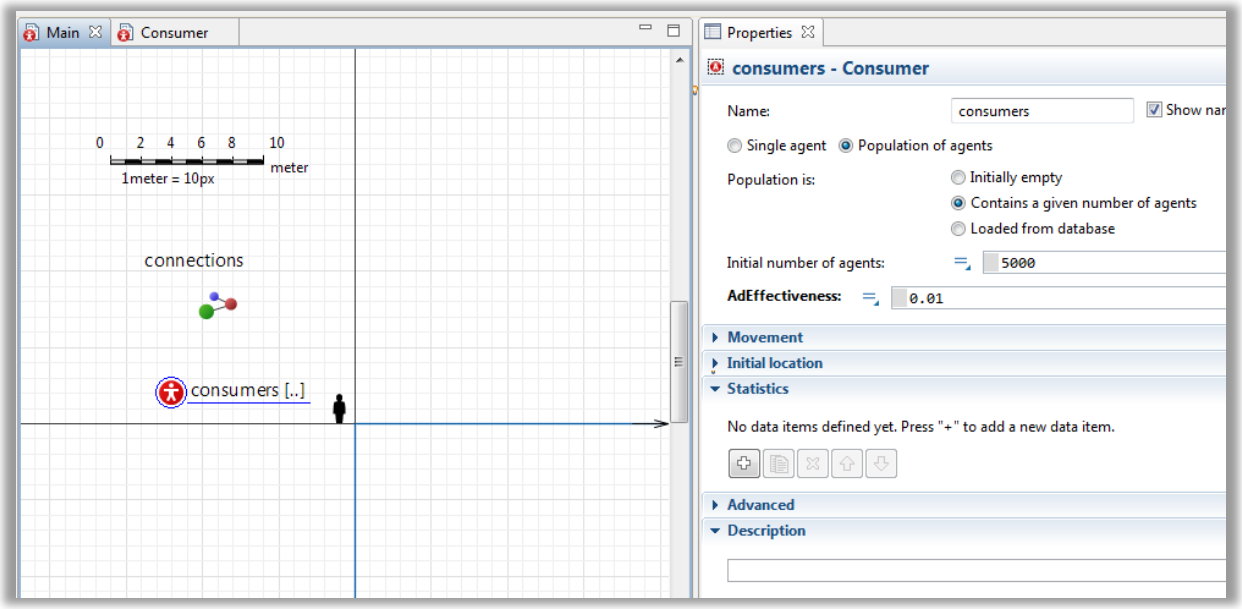

**Résultats d'exécution**



**Ajout d'outils de visualisation**

Il faut sélectionner **consumers** à partir du panel **Main**.

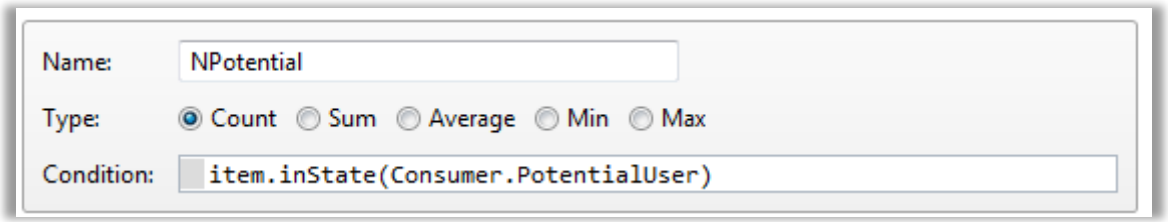
On sélectionne ainsi la classe **consumers**.



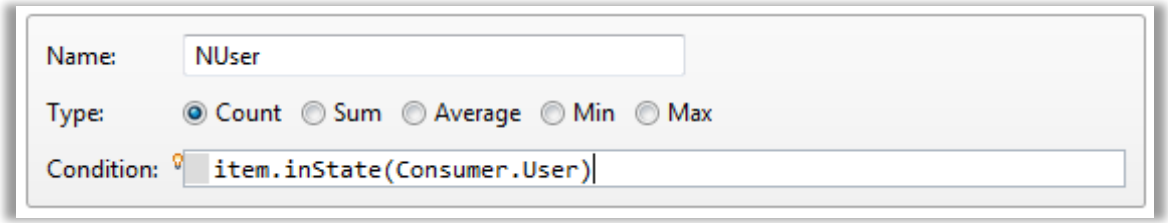

Il faut s'intéresser à l'onglet Statistics.

En utilisant  on peut définir des nouveaux attributs (**qui sont en réalité des méthodes**) pour la classe **Consumer** qui représente la liste de tous les **consumers**.

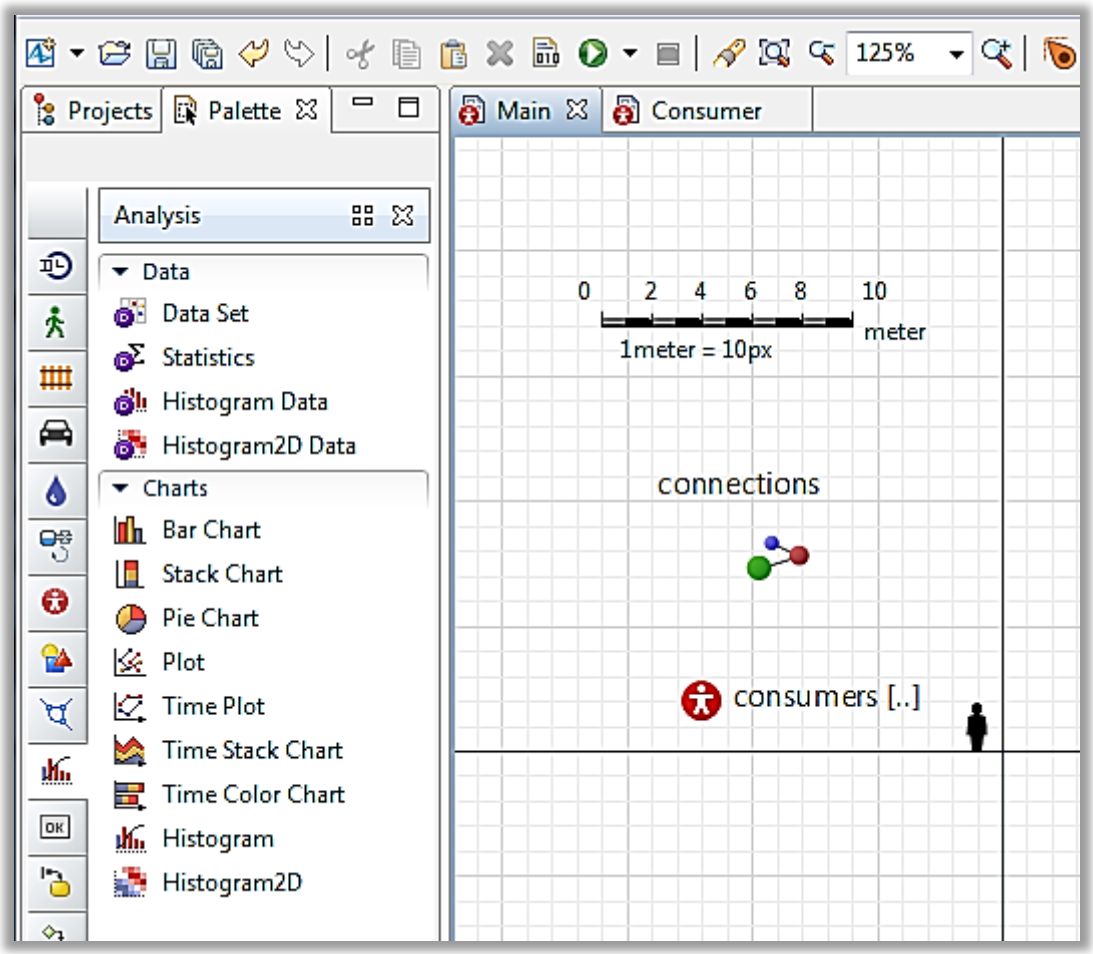
On compte l'ensemble de tous les consumer qui sont dans l'état PotentialUser.



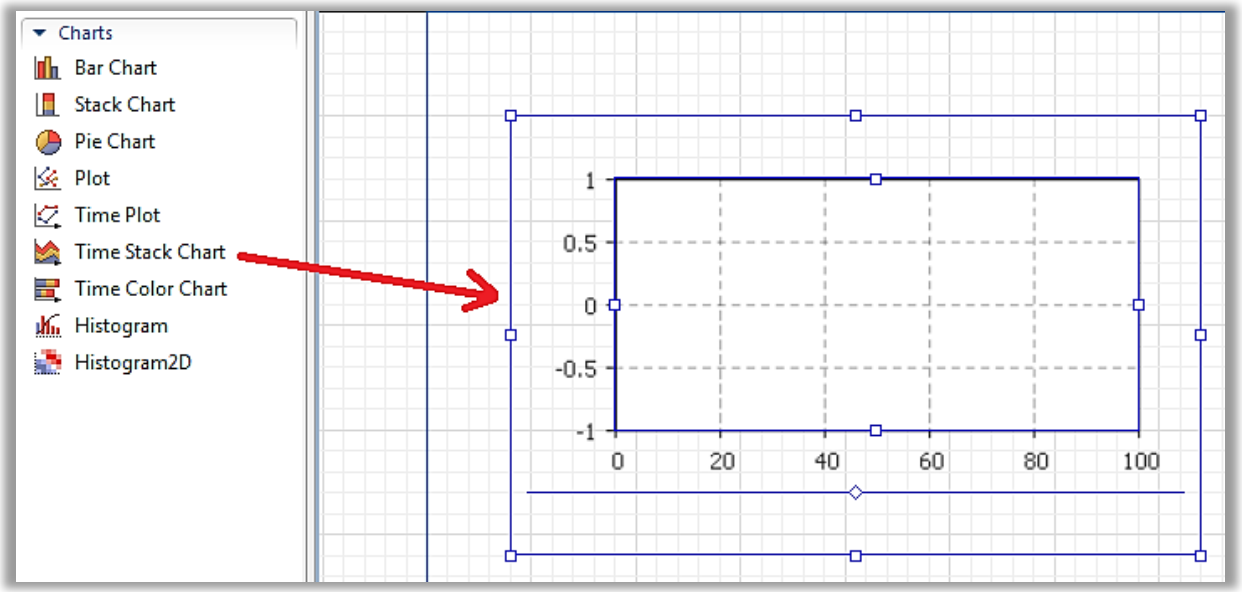

On peut définir de la même manière **NUser** pour compter le nombre de consumer dans l'état **User**.



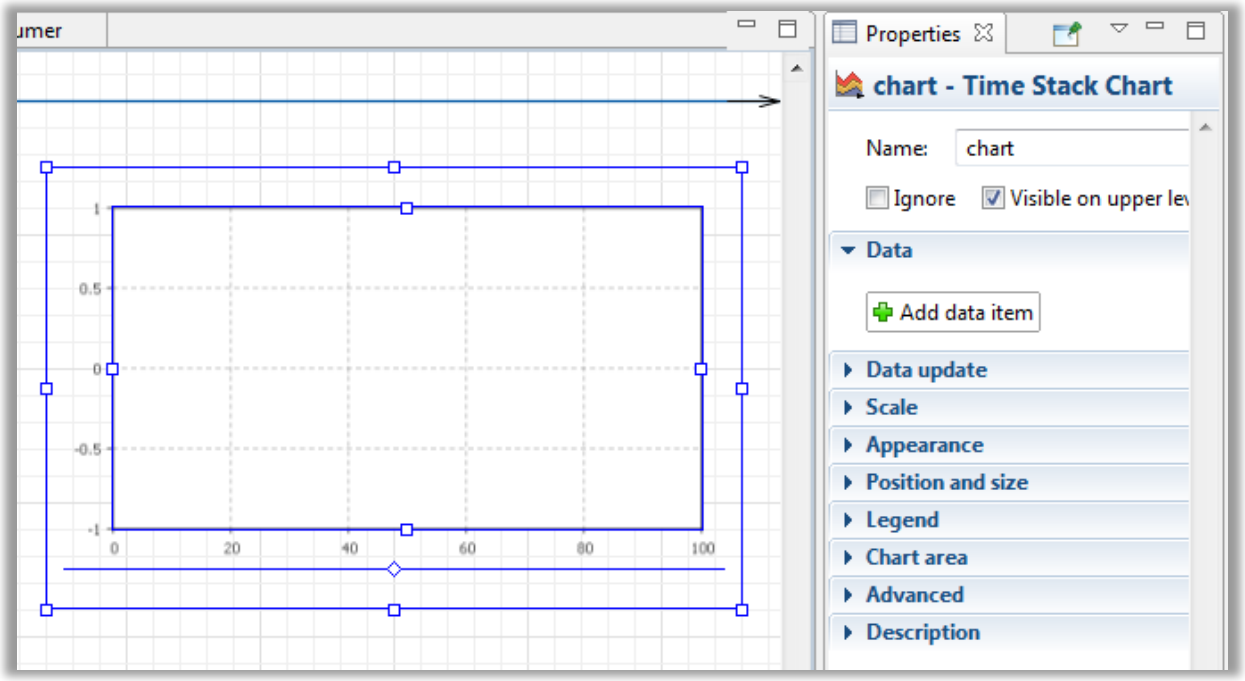
L'ensemble des outils pour créer les rapports sont dans l'onglet : 




Il faut déposer un **Time Stack Chart** sur le page principale.



Il faut modifier la partie **Data**.

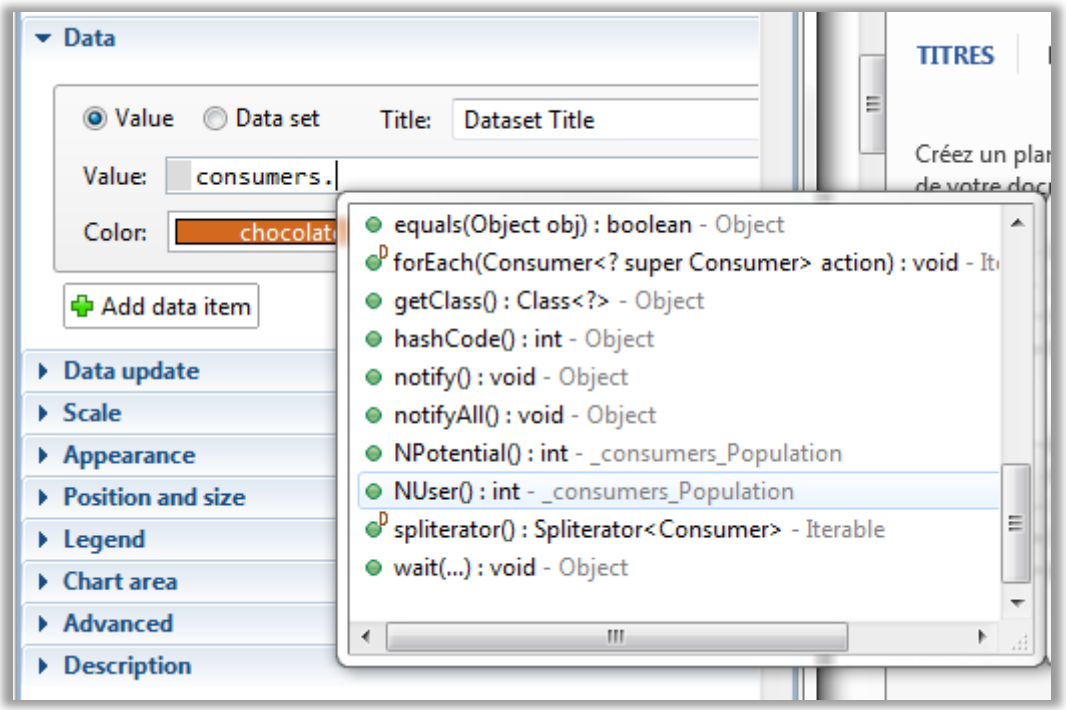



On va représenter sur le schéma deux courbes montrant l'évolution du nombre de consommateurs dans chacun des deux états.

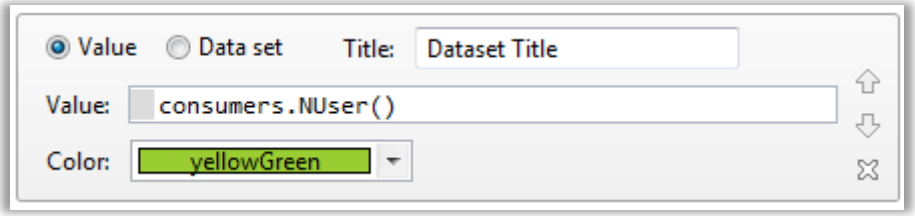
Utilisons le bouton 

pour ajouter des courbes….

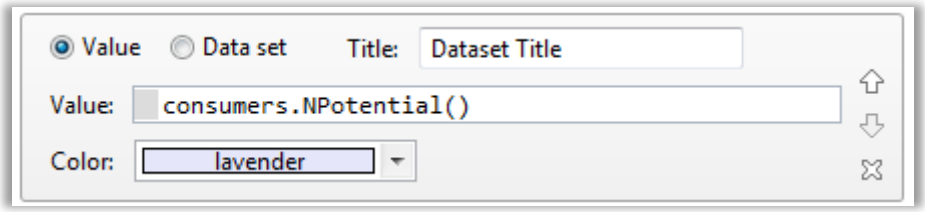
La méthode recherche appartient à **consumers** et on peut utiliser le système de complétion automatique via CTRL + ESPACE.



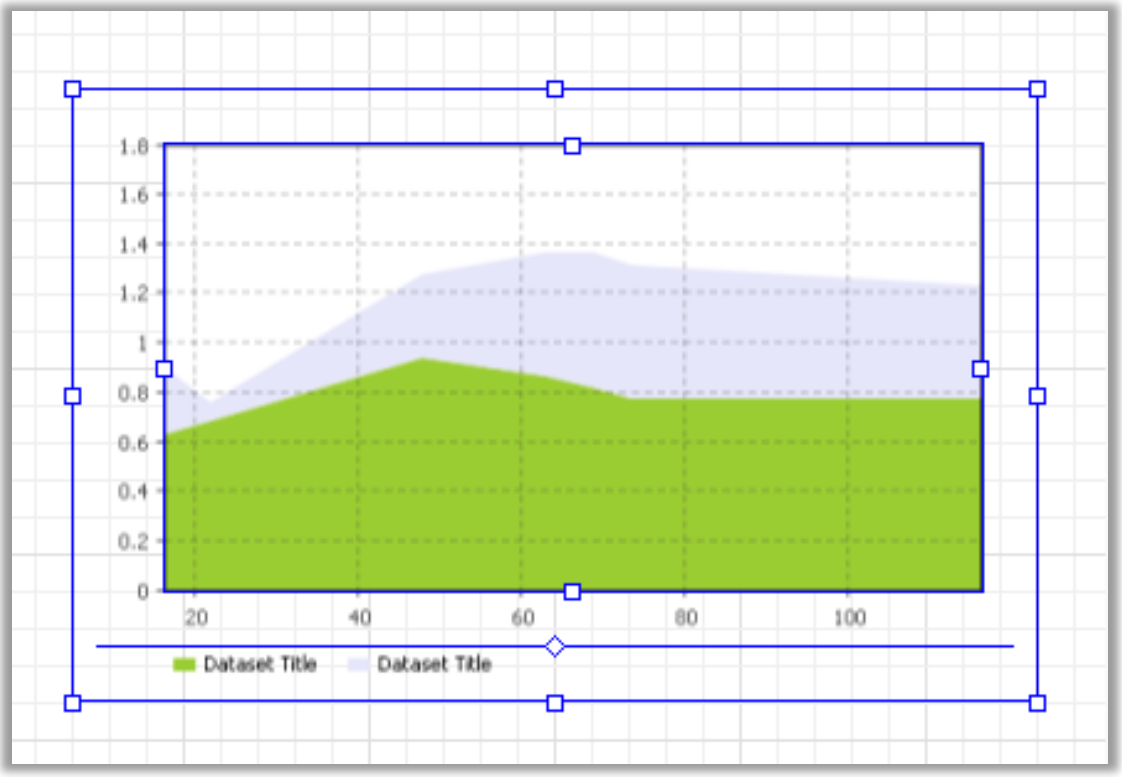
On peut alors choisir une couleur pour la visualisation et comme on a sélectionné le vers pour diagramme d'état, on peut faire le même choix pour être cohérent.



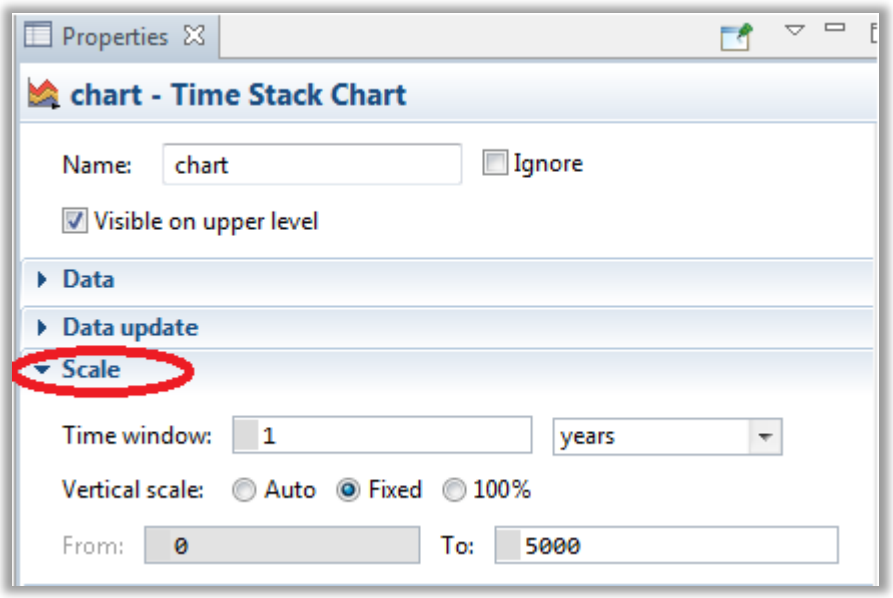

On peut recommencer pour le nombre de consommateurs dans l'état Potential…



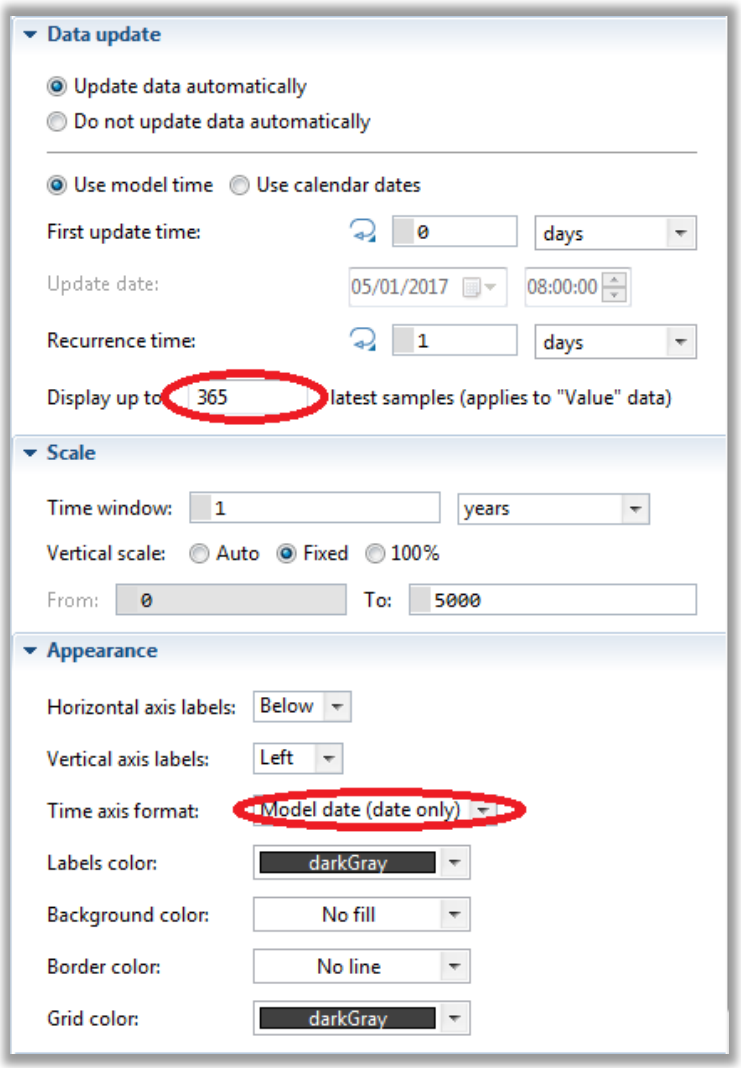

Sur la page en cours de construction, les courbes se présentent comme suit :



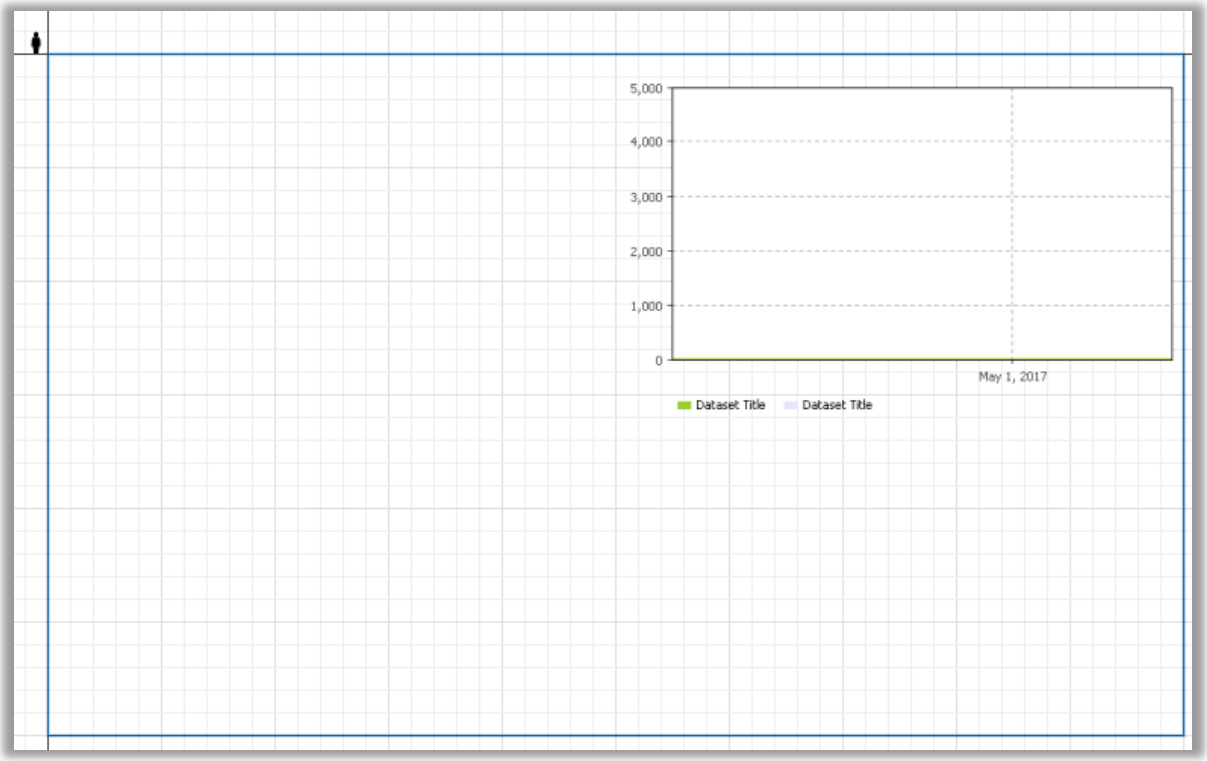

Dans l'onglet Scale, on peut ensuite contrôler l'amplitude de la courbe.
Ici on choisit une période de 1 an avec une amplitude en ordonnées de 5 000.



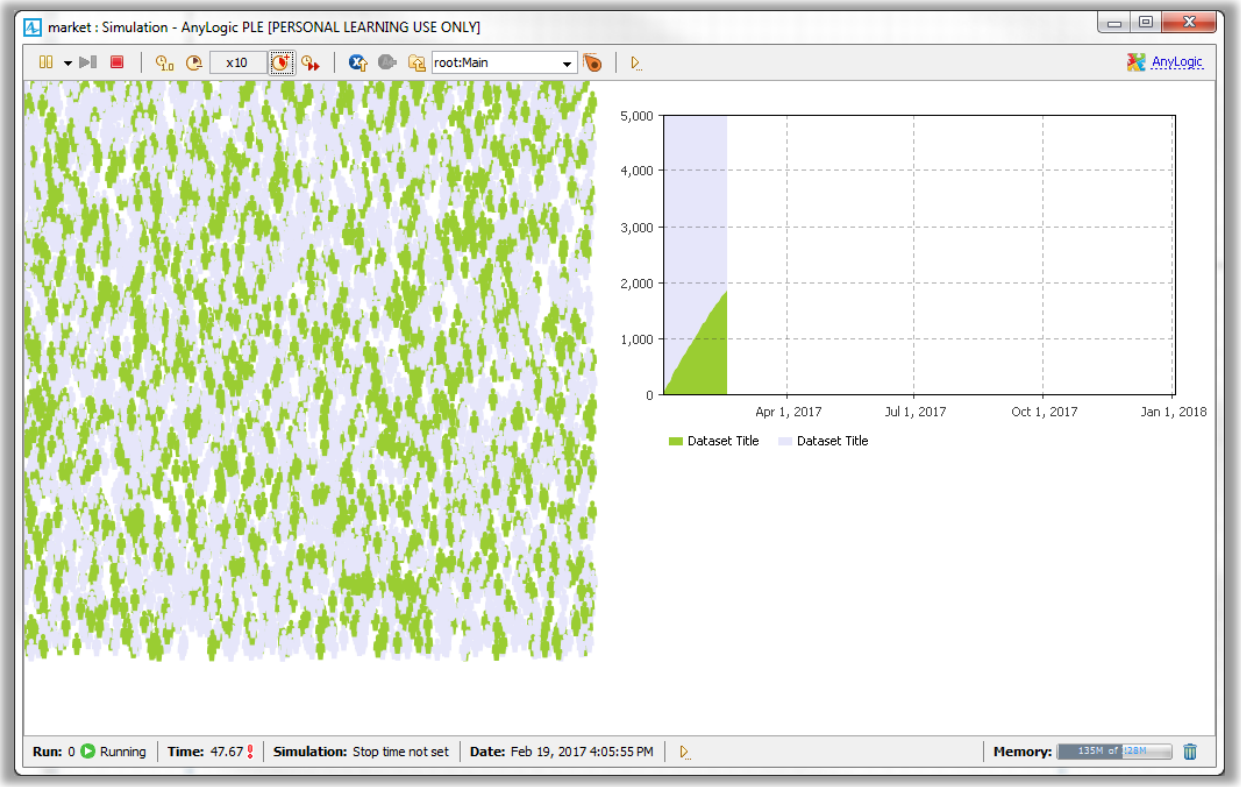

Le rafraichissement de la courbe peut lui aussi être paramétré via l'onglet **Data update** et **Appearance**.



Il faut penser à positionner les courbes à côté et pas sur le modèle... par exemple, en décalant les courbes en haut à droite.




**Résultat d'exécution**

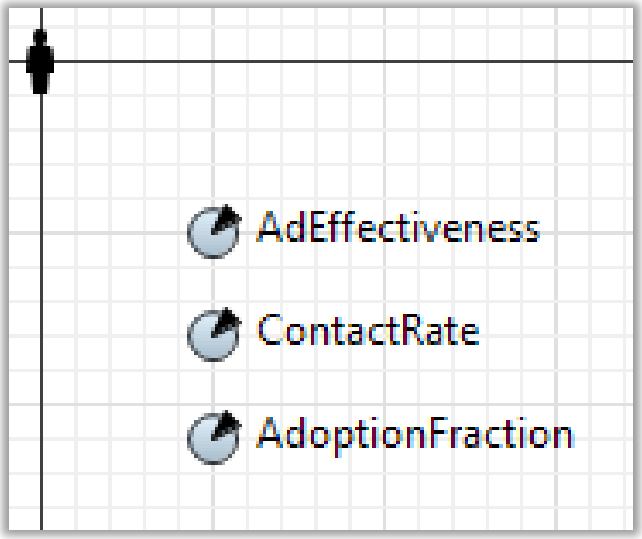


### 3.3. Définir un comportement plus complexe du « consommateur » : notion d'interaction

On va ajouter dans le modèle des interactions entre les agents. Par rapport à ce qu'on fait classiquement en simulation à évènements discrets, on peut considérer qu'il s'agit d'envoi de messages entre des éléments de flux.

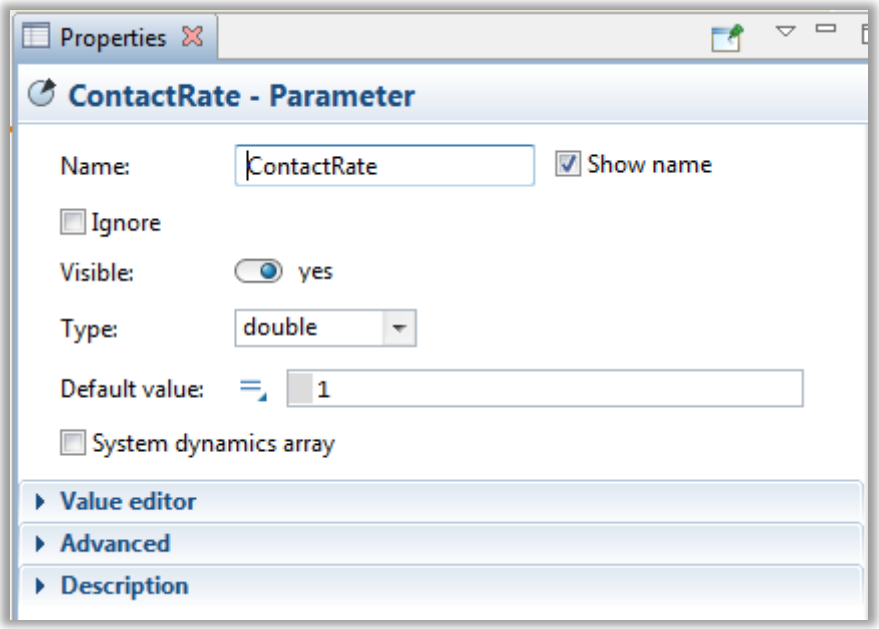
Etape 1. Ajout de deux nouvelles variables globales

On ajoute **ContactRate** et **AdoptionFraction**.

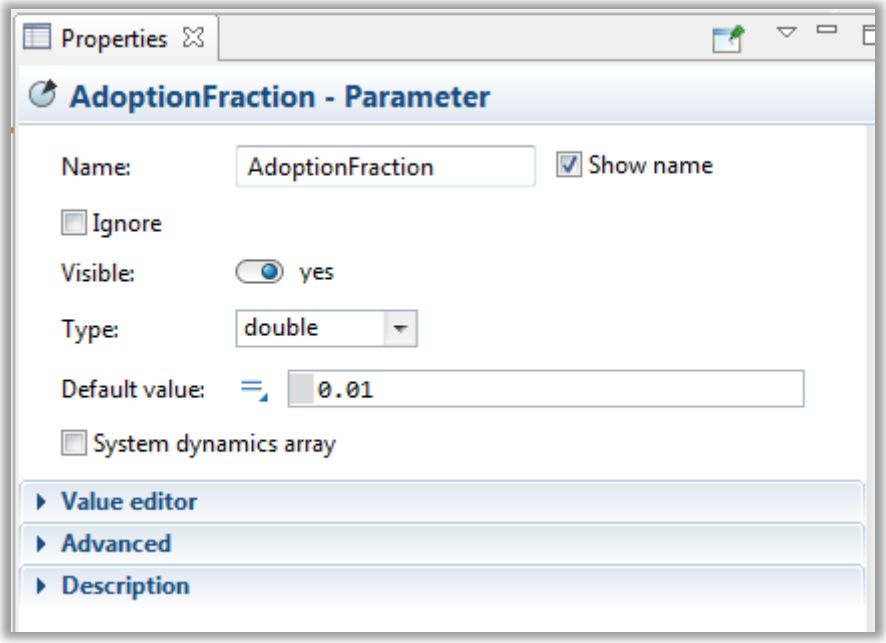



On peut maintenant attribuer des valeurs par défaut.

Par exemple un **ContactRate** de 1 contact par jour.

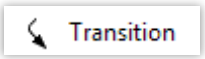


Pour le taux de passage dans l'état User sur la réception d'un message, on définir 0.01 comme valeur (1% de chance).

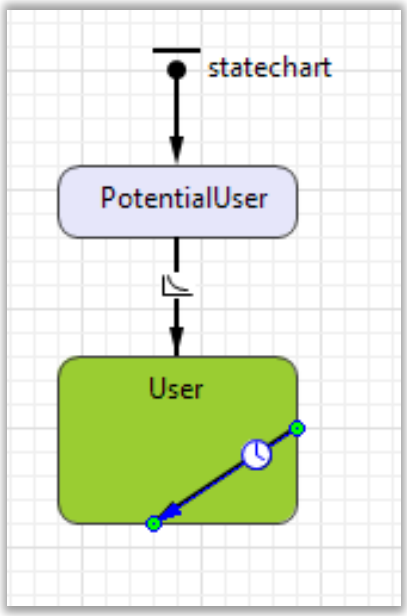



Etape 2. Modification du schéma

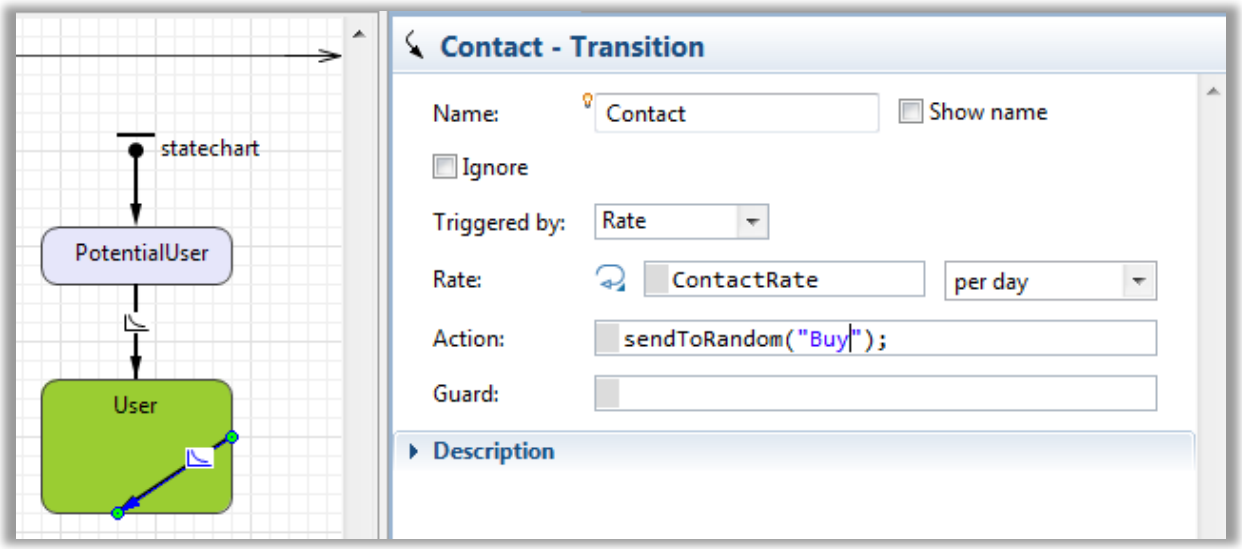
Pour cela on va ajouter une **transition interne** à un état : ici l'état **USER**.

Pour cela il faut utiliser : 


Ce type de transition définit une opération qui est effectuée de manière répétitive. Visuellement, le schéma se présente comme ceci :

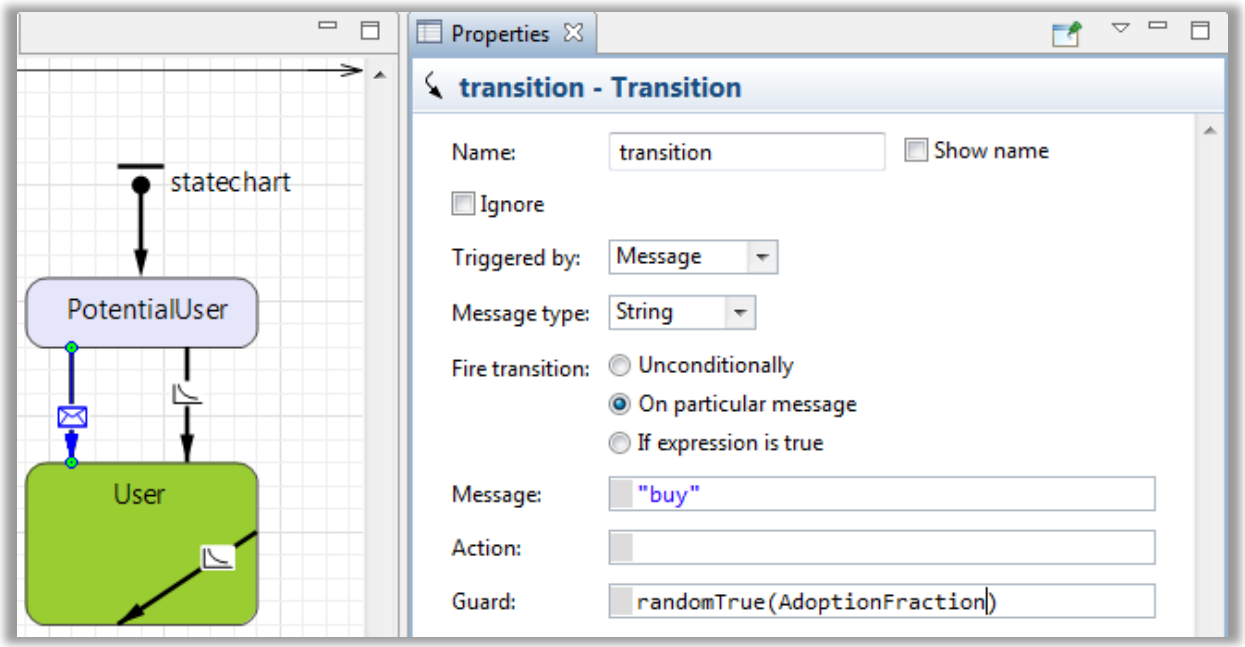



Ceci signifie que périodiquement, selon le taux **ContactRate**, un agent émet un message **Buy** à destination d'un agent choisit aléatoirement.

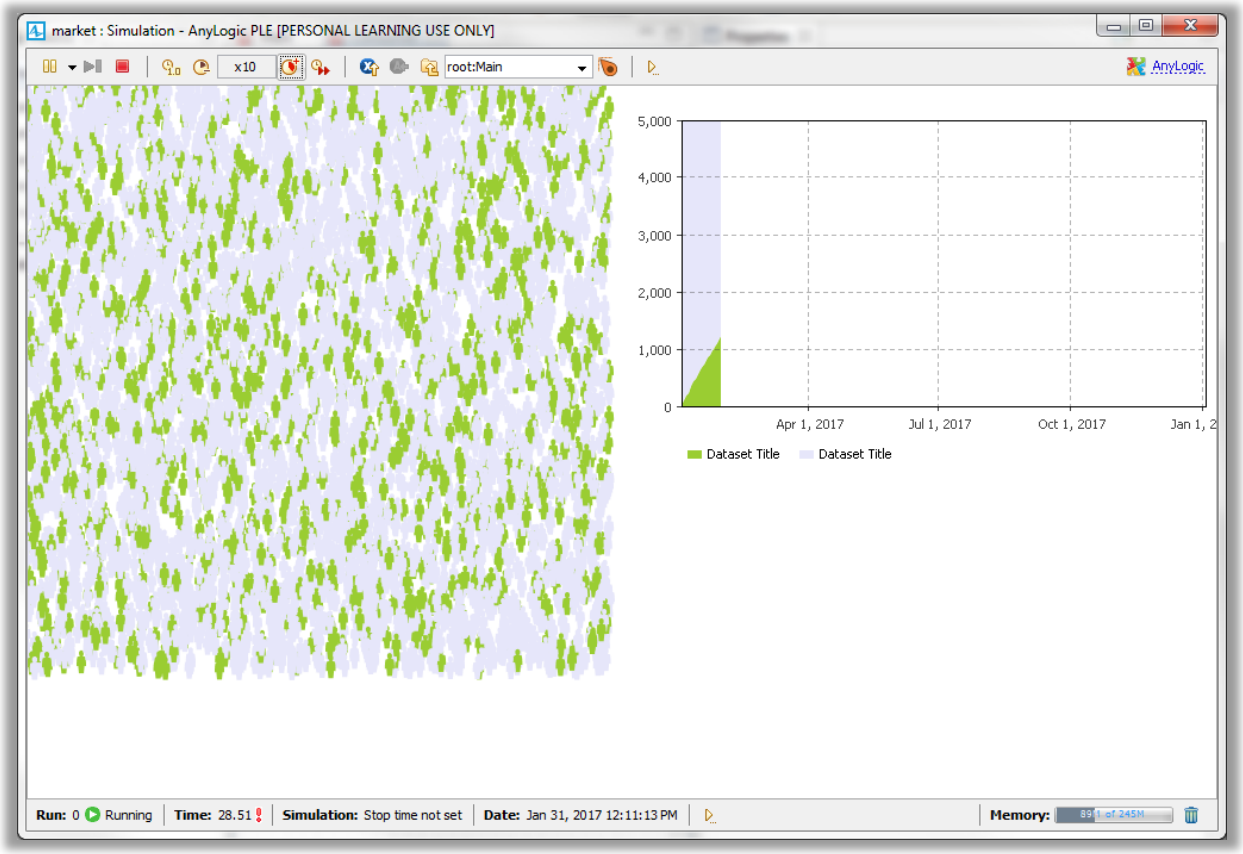



Un agent recevant le message « **buy** » va passer, dans certaines conditions dans l'état **User**.
Il faut dont créer :

- Une transition déclenchée sur la réception d'un Message ;
- Spécifier le type de message, ici String ;
- Spécifier le contenu « buy » ;
- et définir une condition supplémentaire (ici on accepte la transition avec une certaine probabilité).



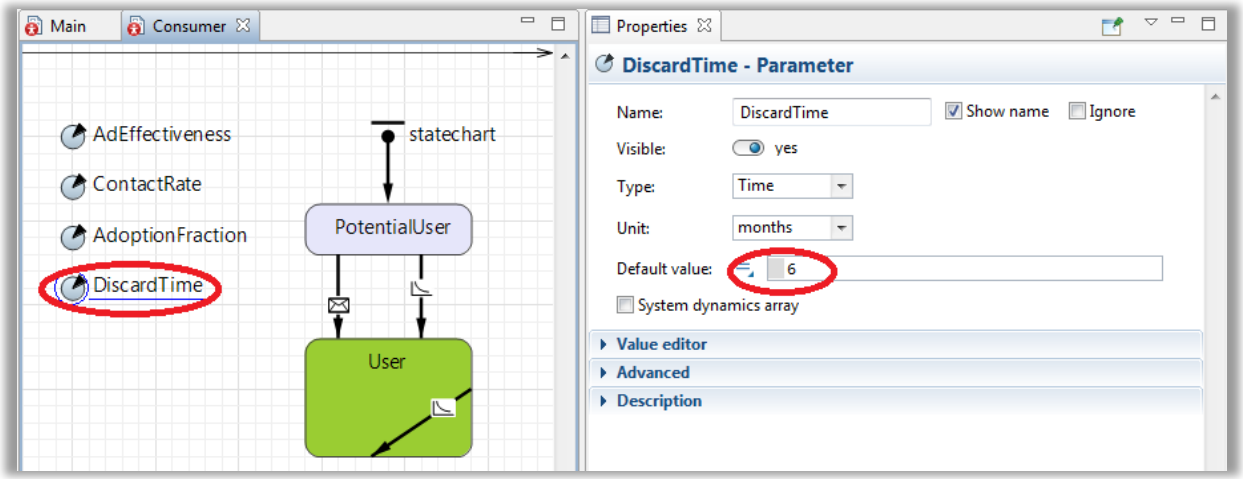
Etape 3. Vérifier l'exécution du modèle




**3.4. Définir un comportement plus complexe du « consommateur » : les produits sont périssables et doivent être renouvelés.**

Etape 1. Ajout d'une nouvelle variable globale

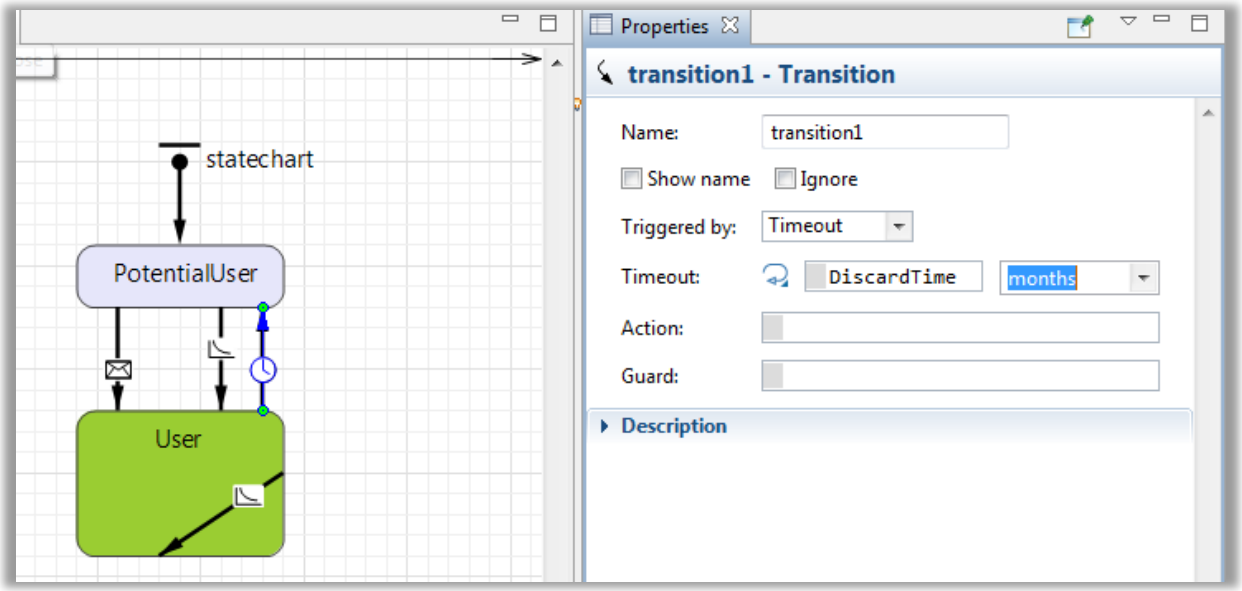
La variable a une valeur de 6 mois.



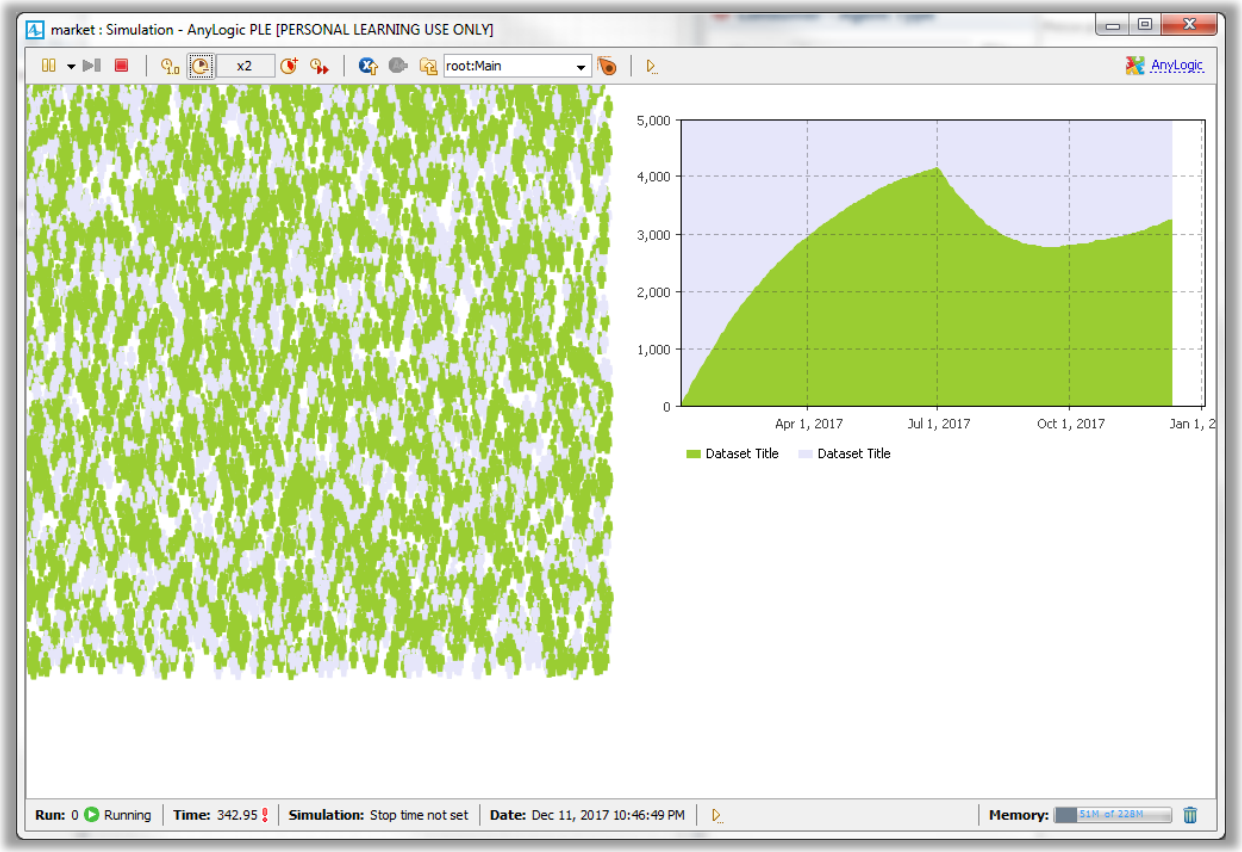

Ce qu'on va exprimer c'est qu'au bout de 6 mois, le consommateur repasse dans l'état **PotentialUser**.

## Etape 2. Ajout d'une nouvelle transition

La transition est validée sur une condition temporelle de valeur **DiscardTime**.



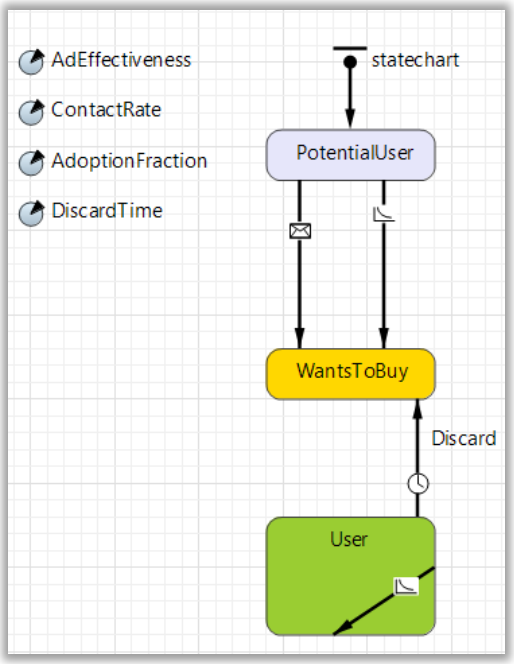

## Etape 3. Résultat d'exécution

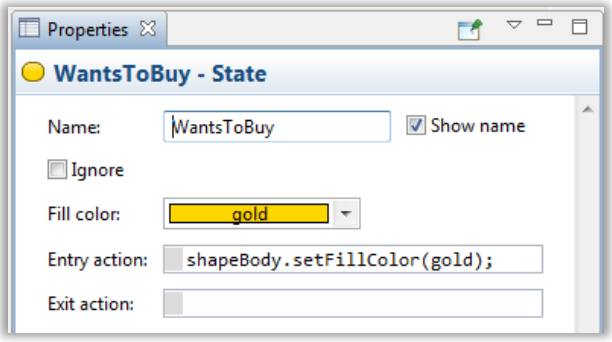


### 3.4. Définir un comportement plus complexe du « consommateur » : prise en compte d'un délai de livraison

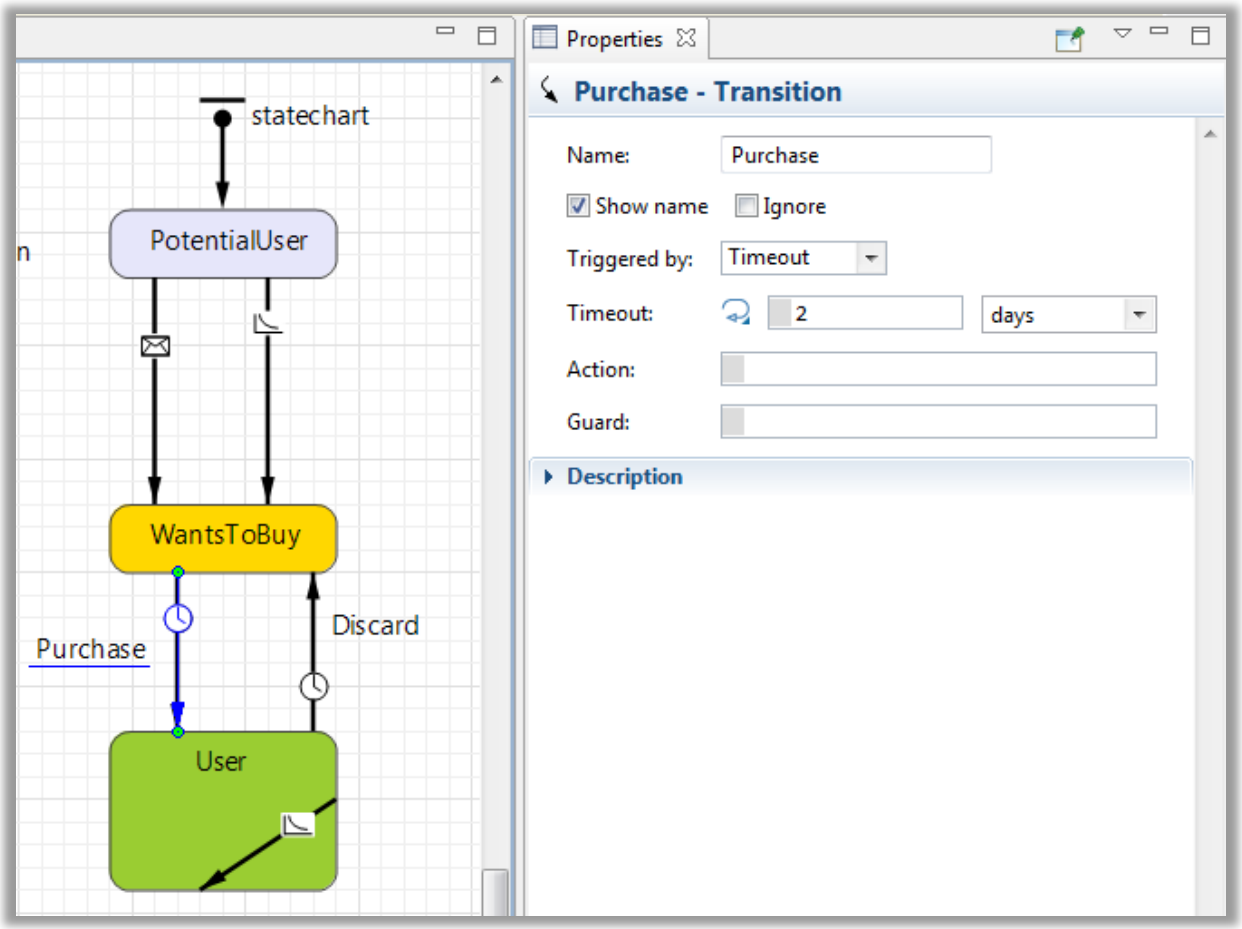
Etape 1. Ajout d'un nouvel état

Il faut modifier le schéma et inclure un nouvel état **WantsToBuy**.

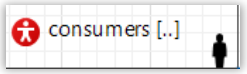


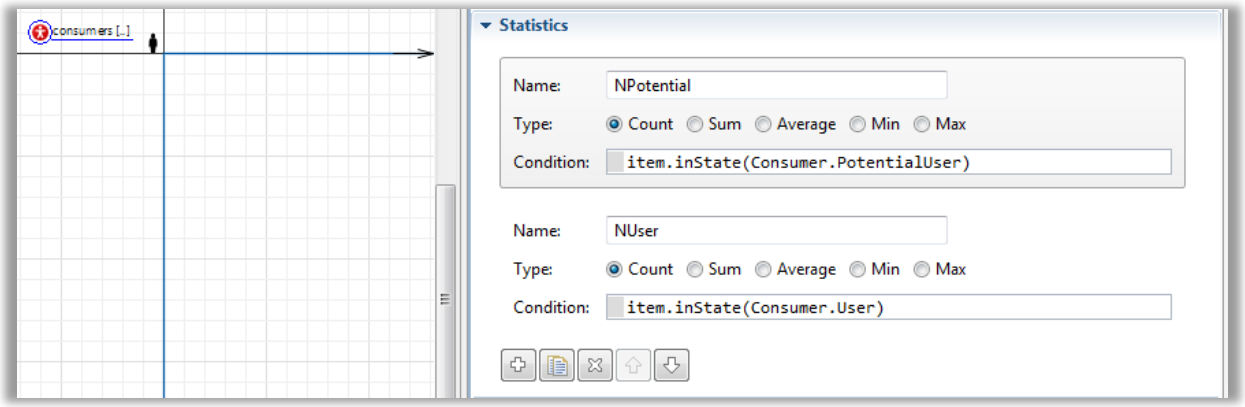

Il faut ajouter un passage de l'état **WantsToBuy** à celui de **Users** comme ci-dessous.



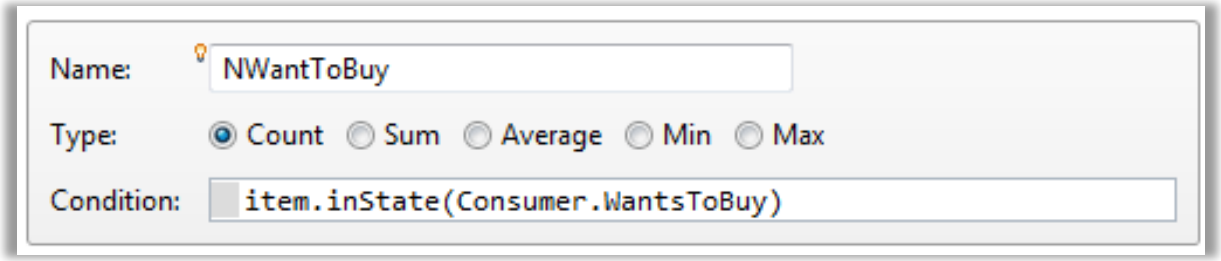
## Etape 2. Ajout de nouvelles statistiques

Il faut sélectionner 

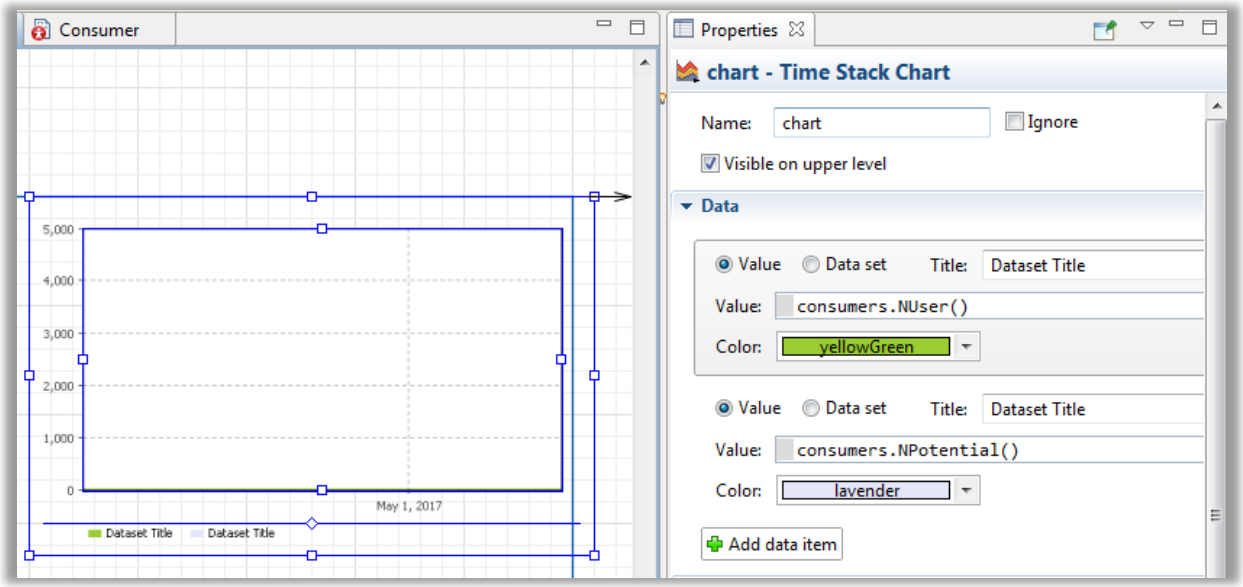
pour ajouter une nouvelle méthode.
Actuellement 2 méthodes figurent dans la partie **Statistics**.



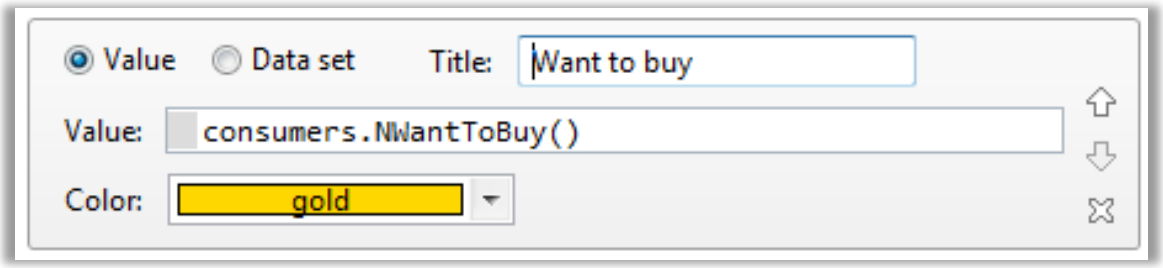

La méthode est nommée **NWantToBuy**.



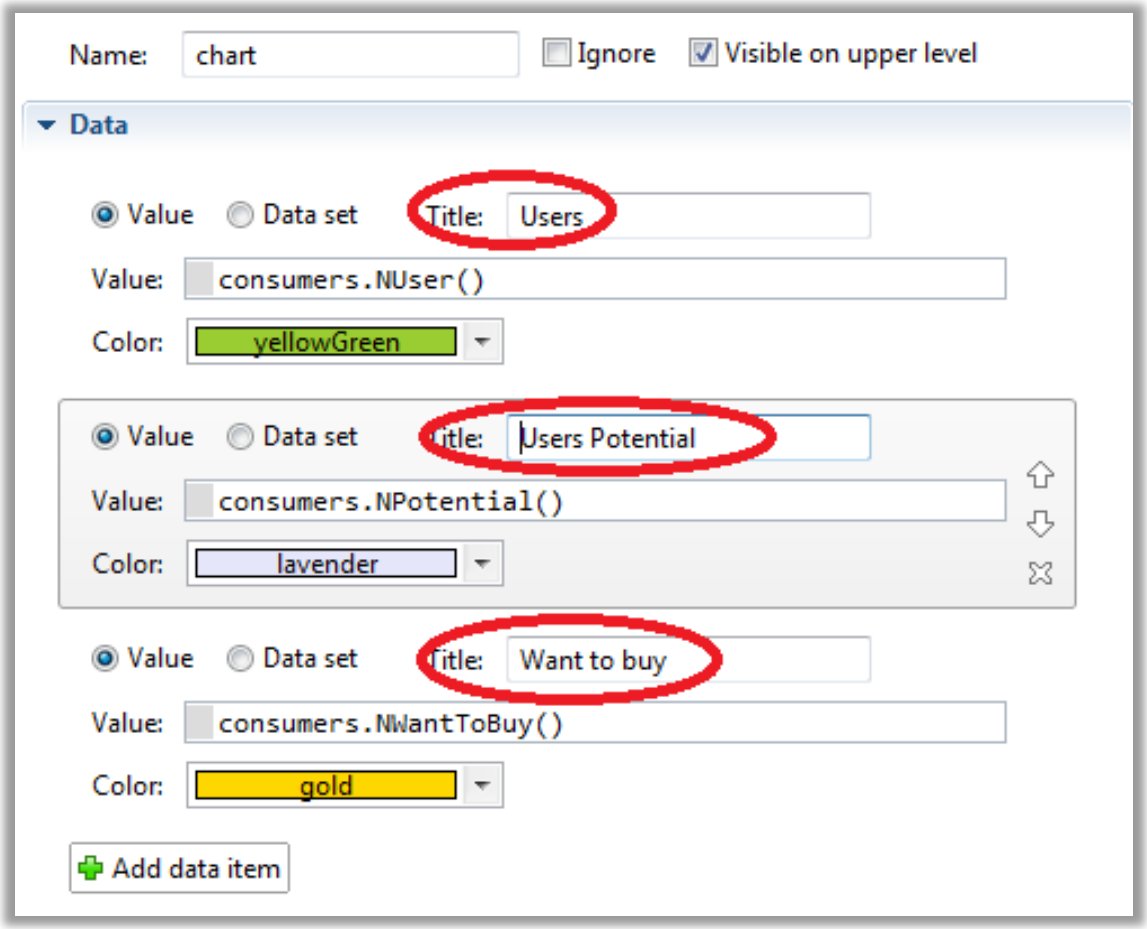

Seules deux courbes sont utilisées sur le schéma. Pour les consulter, il faut « retrouver » le dessin et consulter l'onglet **Data**.



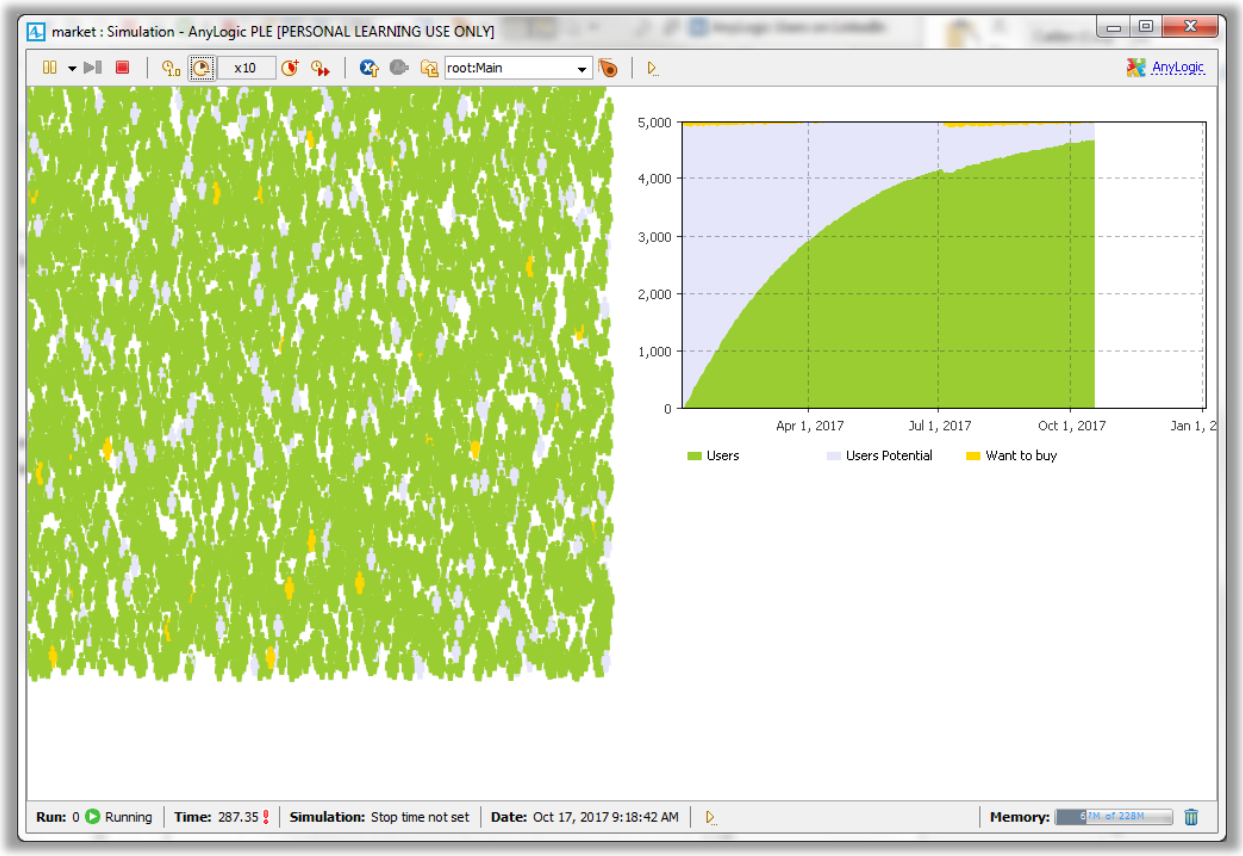
On ajoute une courbe représentant le nombre de consommateurs… « en attente » d'un achat….




Pour augmenter la lisibilité, on peut en profiter pour modifier le champ **Title** des différentes courbes.



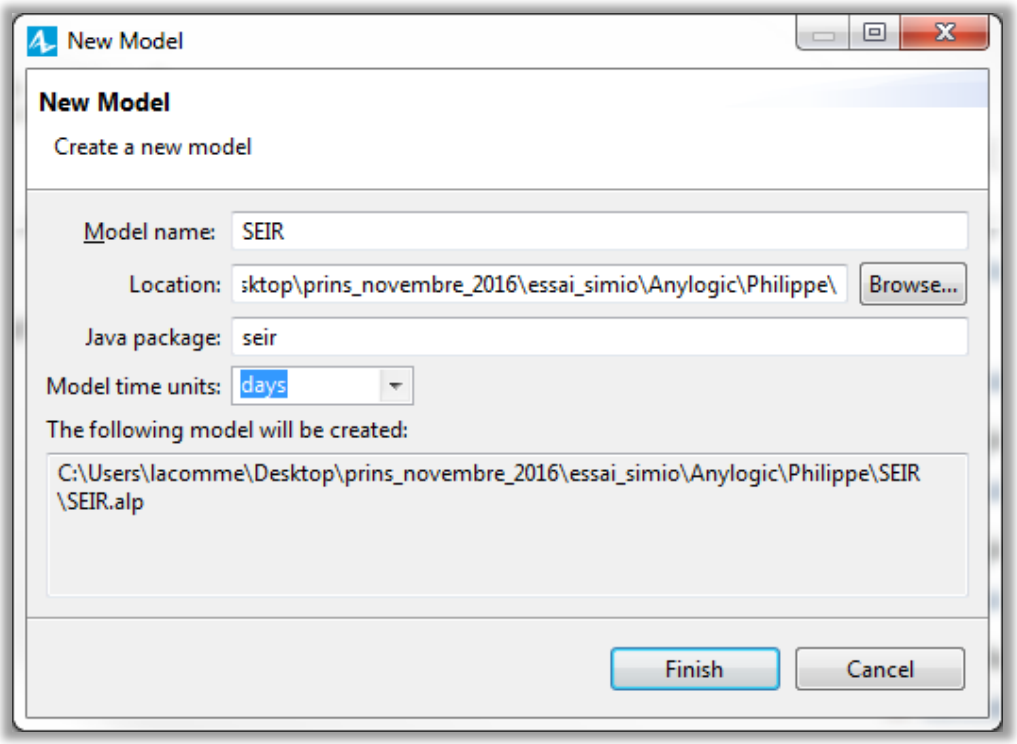
Etape 3. Résultat d'exécution




## 4) Modèle de simulation dynamique : machine à état

### 4.1. Création d'un modèle.

Il faut créer un projet nommé SEIR qui va simuler une épidémie.

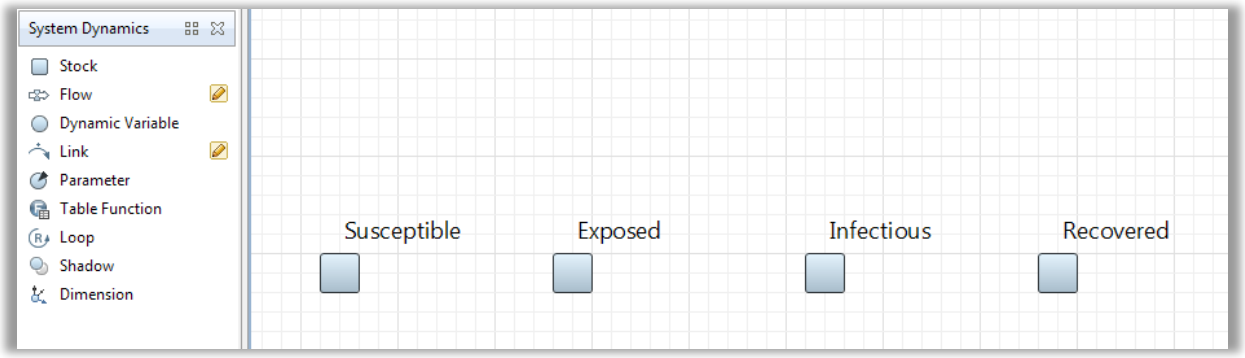



L'ensemble des composants dont nous allons avoir besoin sont dans l'onglet  *i.e.* **System Dynamics**.

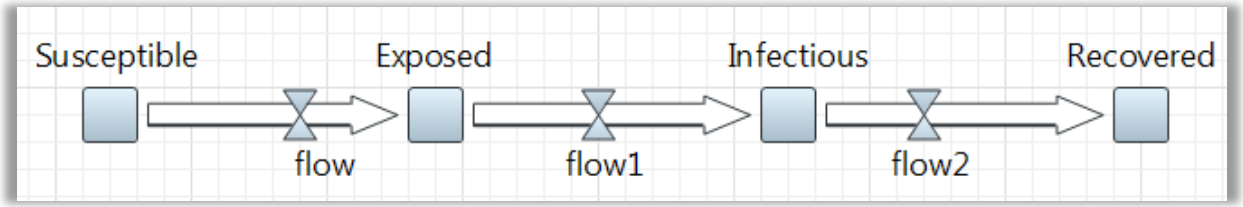
## 4.2. Modélisation

**Etape 1. Définition des états**

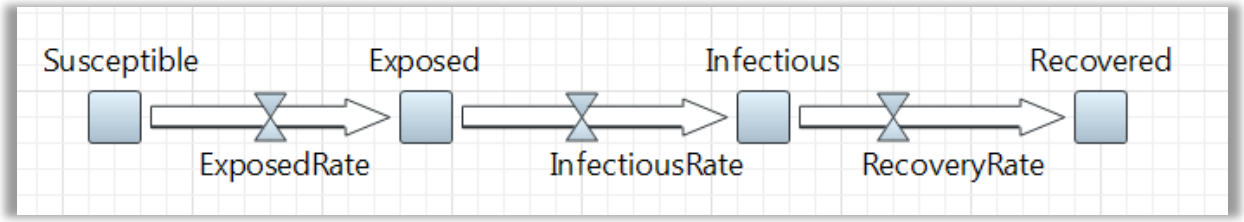
On introduit 4 états à partir du composant **Stock** de l'onglet **System Dynamics**.



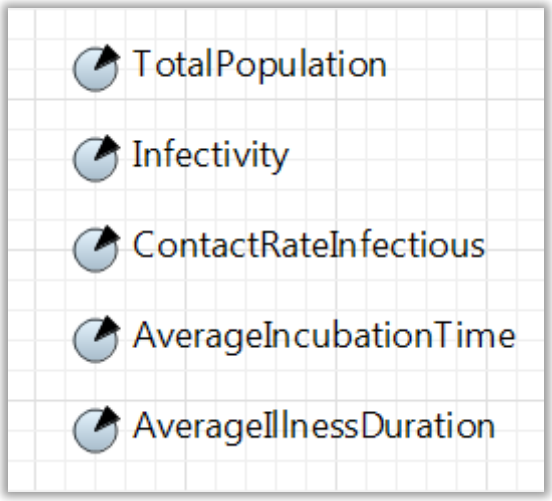

Il faut ajouter ensuite les flots.



On peut ensuite donner un nom aux flots décrivant les changements d'état.




Ajouter ensuite 5 variables globales

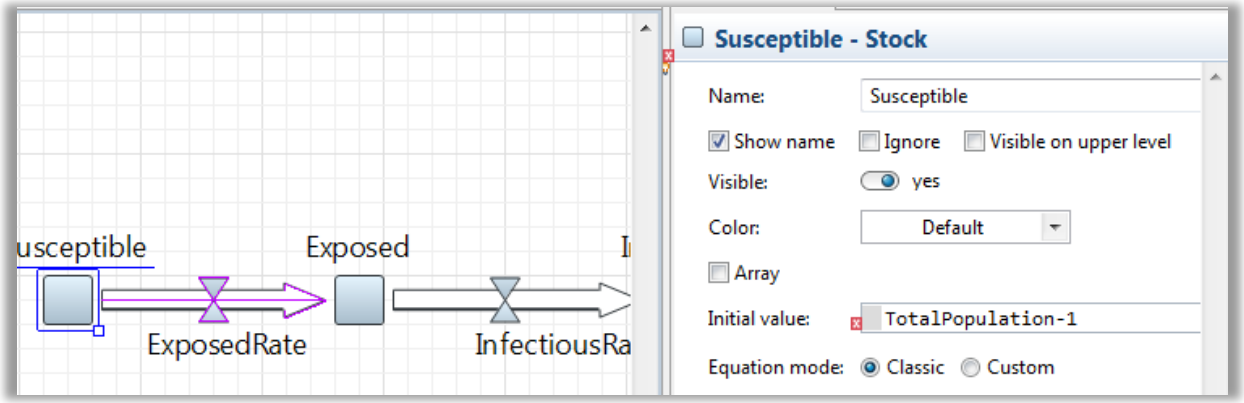



Avec comme valeur :

- TotalPopulation = 10 000
- Infectivity = 0.6 avec le jour comme unité (days)
- ContactRateInfectious = 1.25 avec le jour comme unité (days)
- AverageIncubationTime = 10 avec le jour comme unité (days)
- AverageIllnessDuration = 15 avec le jour comme unité (days)

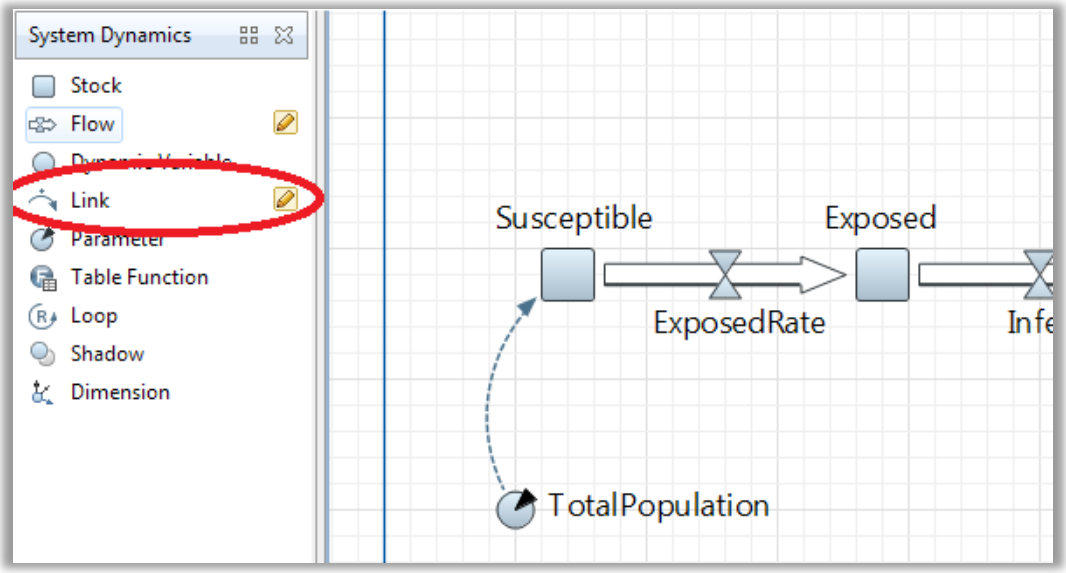
**Etape 2. Définition du stock initial (état : Susceptible)**

On définit ensuite une valeur pour le nombre initial de personnes infectées. Ici 1 personne, ce qui signifie que le nombre de personnes non infectées mais susceptible de l'être est : TotalPopulation-1.

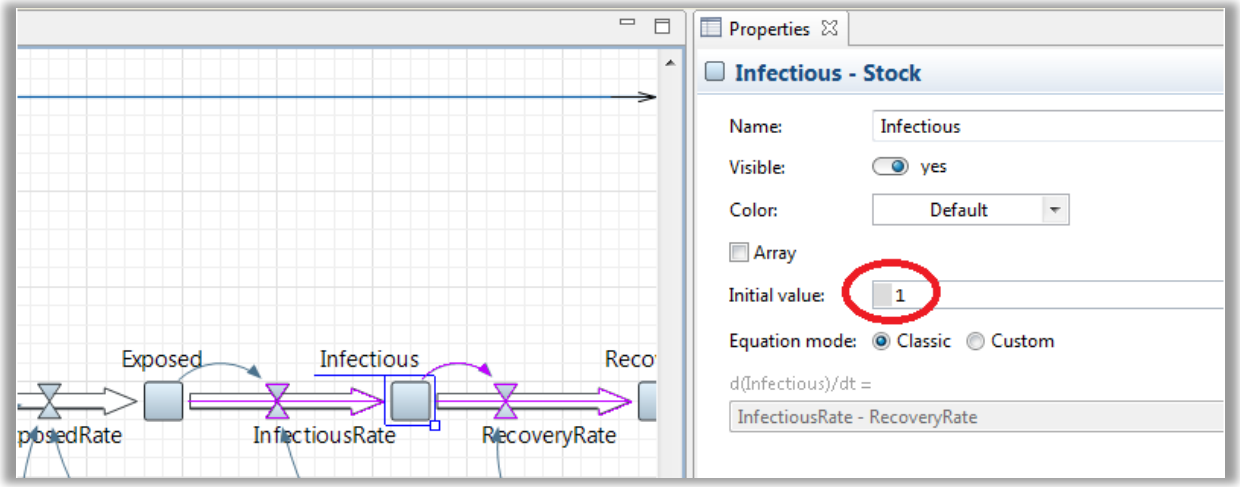



Il faut noter le symbole route à côté de Initial Value qui montre qu'on ne peut pas utiliser la variable TotalPopulation car celle-ci est inconnue au niveau du stock.

Il faut ajouter un lien entre la variable **TotalPopulation** et **Susceptible**.




On suppose qu'un seul individu est dans l'état **Infectious** et on définit 1 comme valeur initiale.

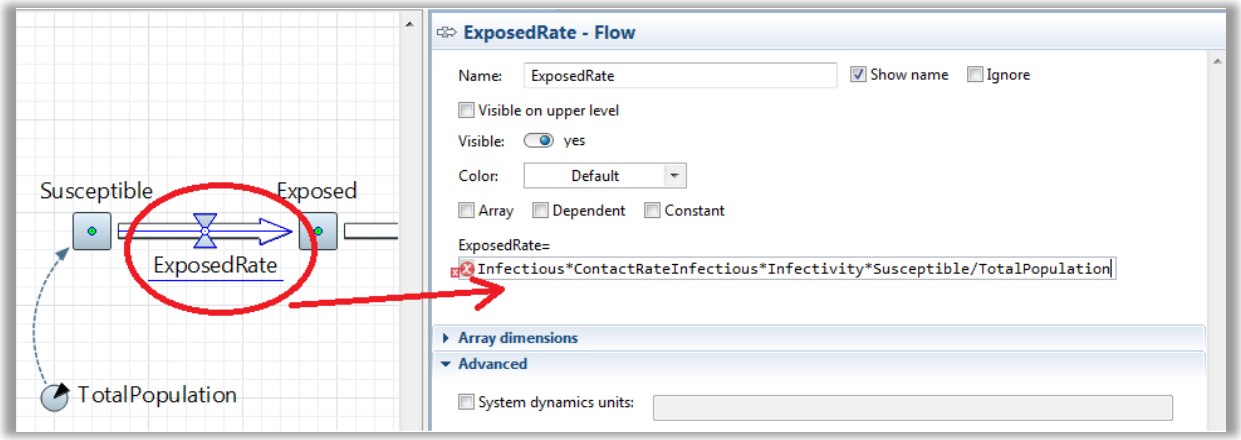



**Etape 3. Passage de Suceptible à exposer.**

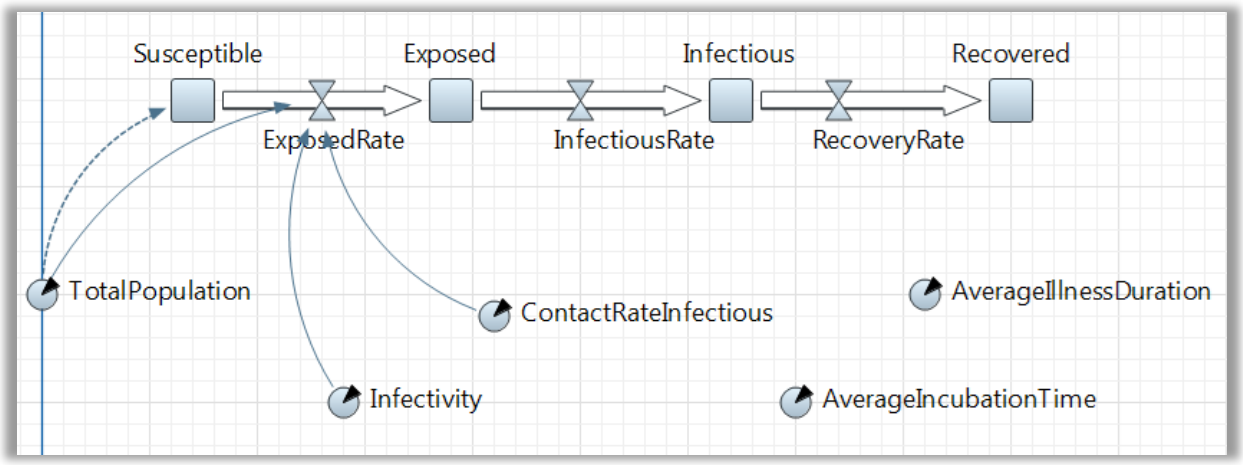
`Le nombre de personnes exposées est défini par :`

`Infectious*ContactRateInfectious*Infectivity*Susceptible/TotalPopulation`

Il faut saisir cette information sur le flow.

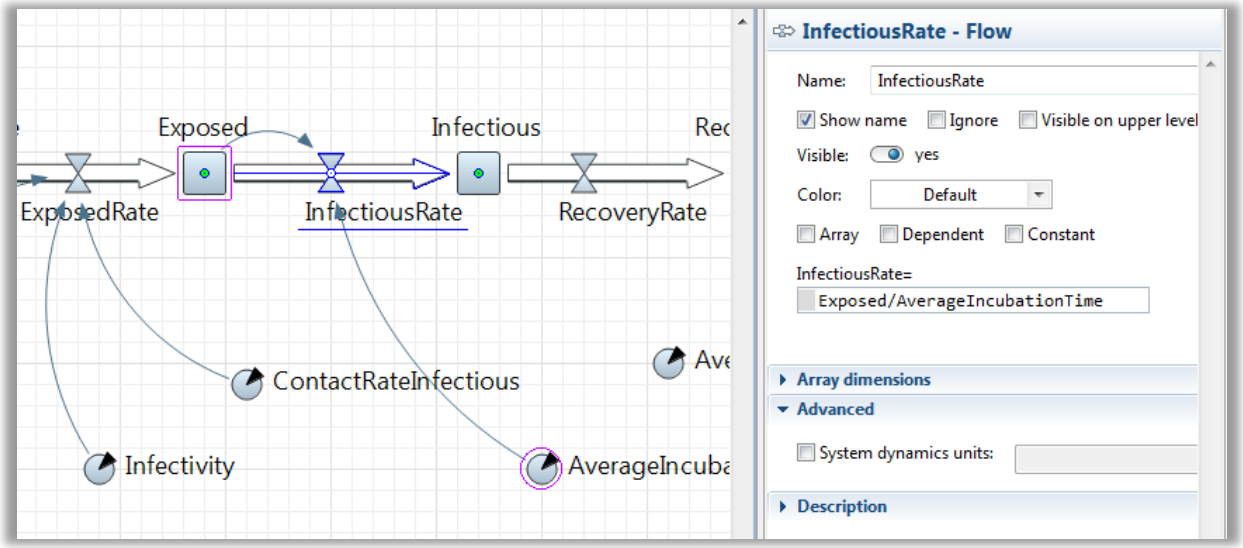


Pour les mêmes raisons que précédemment, il faut relier les variables globales au flot pour déclarer leur utilisation.



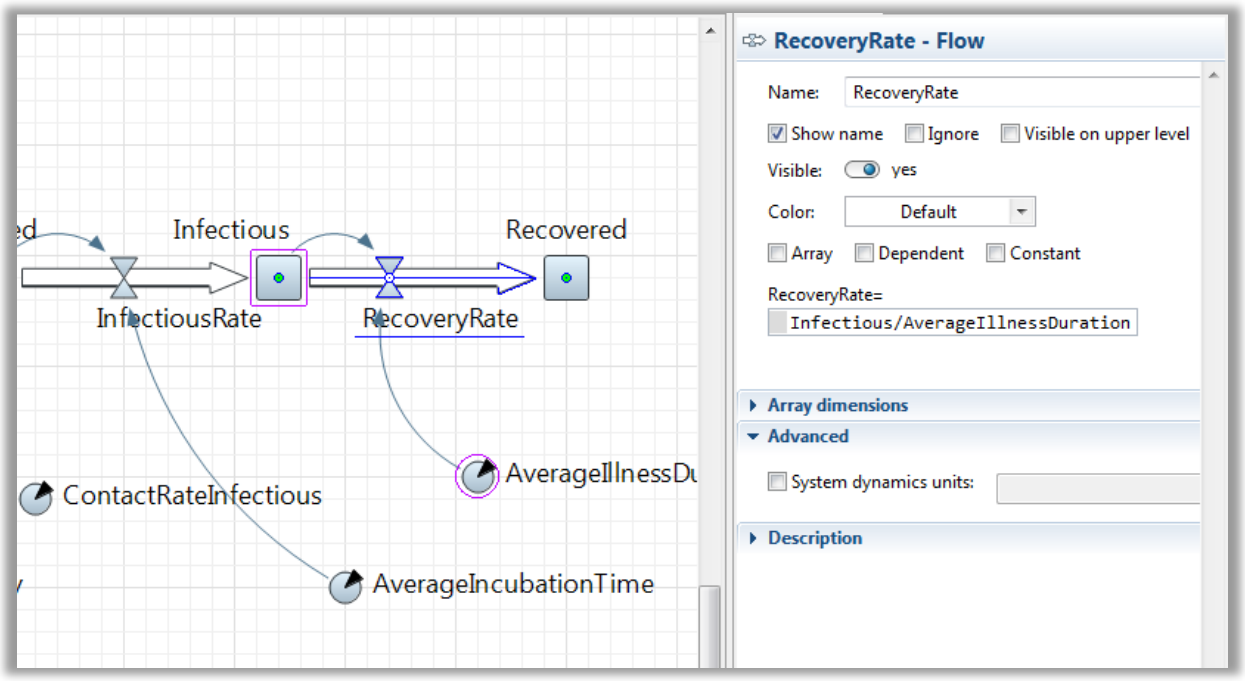

**Etape 4. Passage d'Exposed à Infectious**

Le nombre de personnes infectées est donné par le nombre de personnes exposés divisé par le temps moyen d'incubation : `Exposed/AverageIncubationTime`

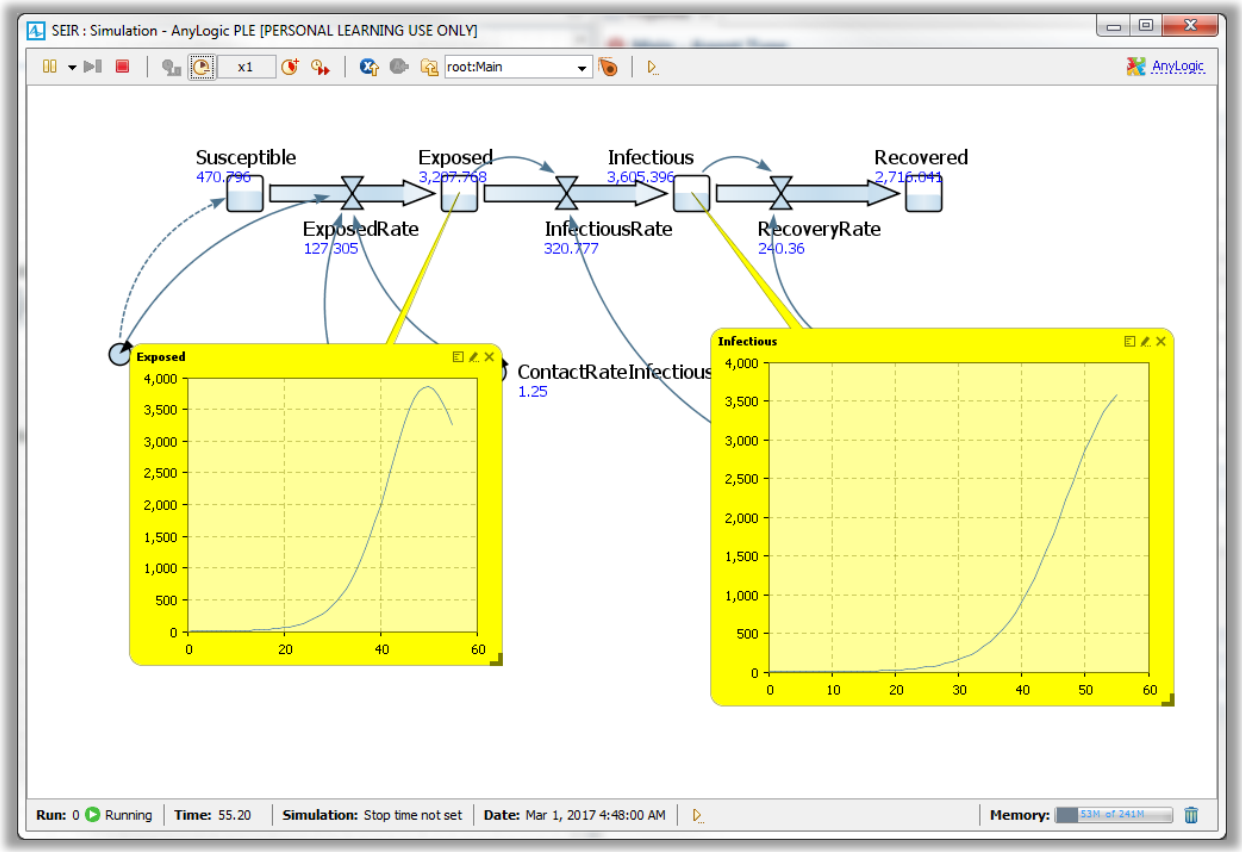



**Etape 5. Passage de Infectious à Recovered**

Le nombre de personne ayant retrouvé la santé, est défini par le nombre de personne malade/la durée moyen de guérison : `Infectious/AverageIllnessDuration`

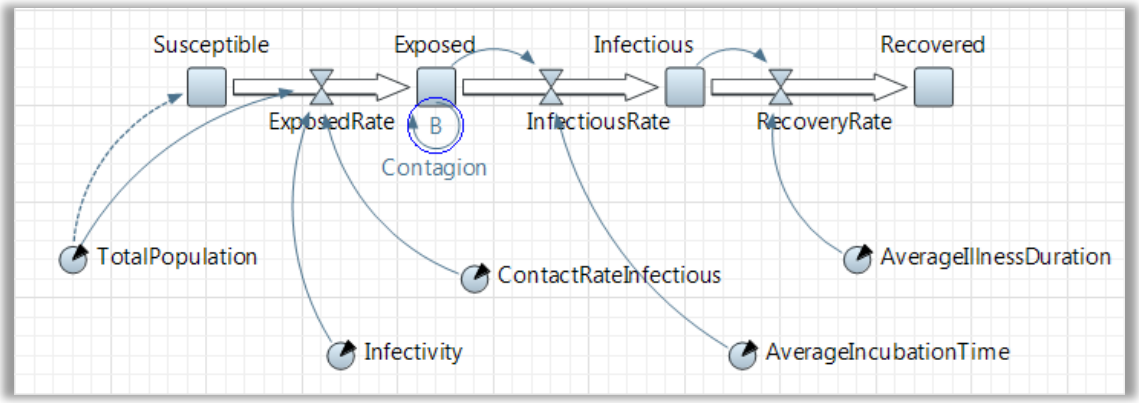



## 4.3. Exécution du modèle

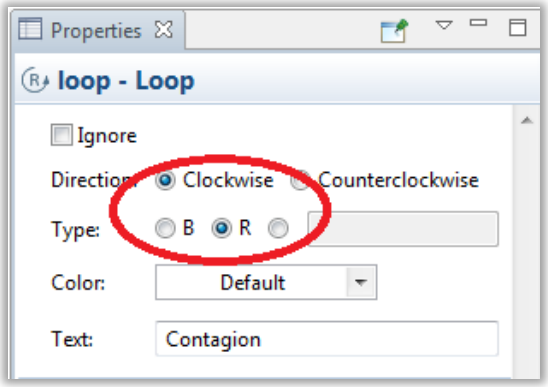


### 4.4. Visualisation de la dynamique du système

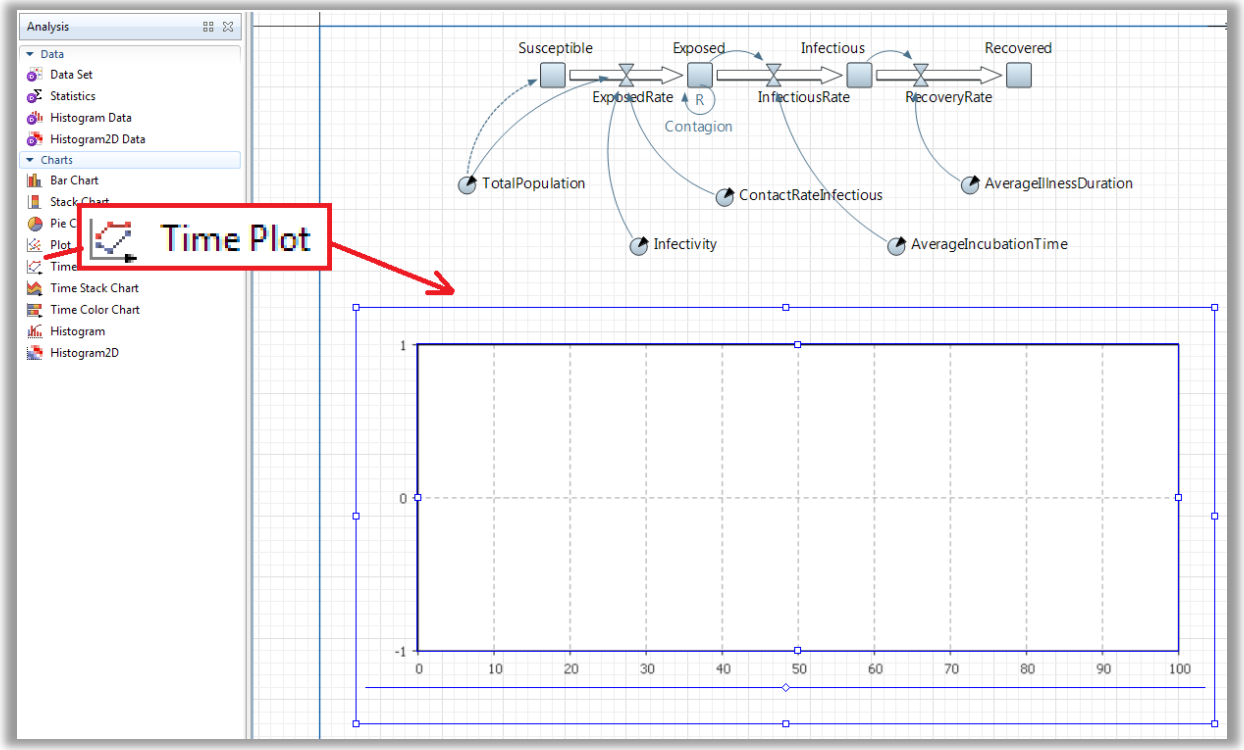
Pour mieux appréhender la dynamique, on ajouter une boucle de renforcement nommée **Contagion**.



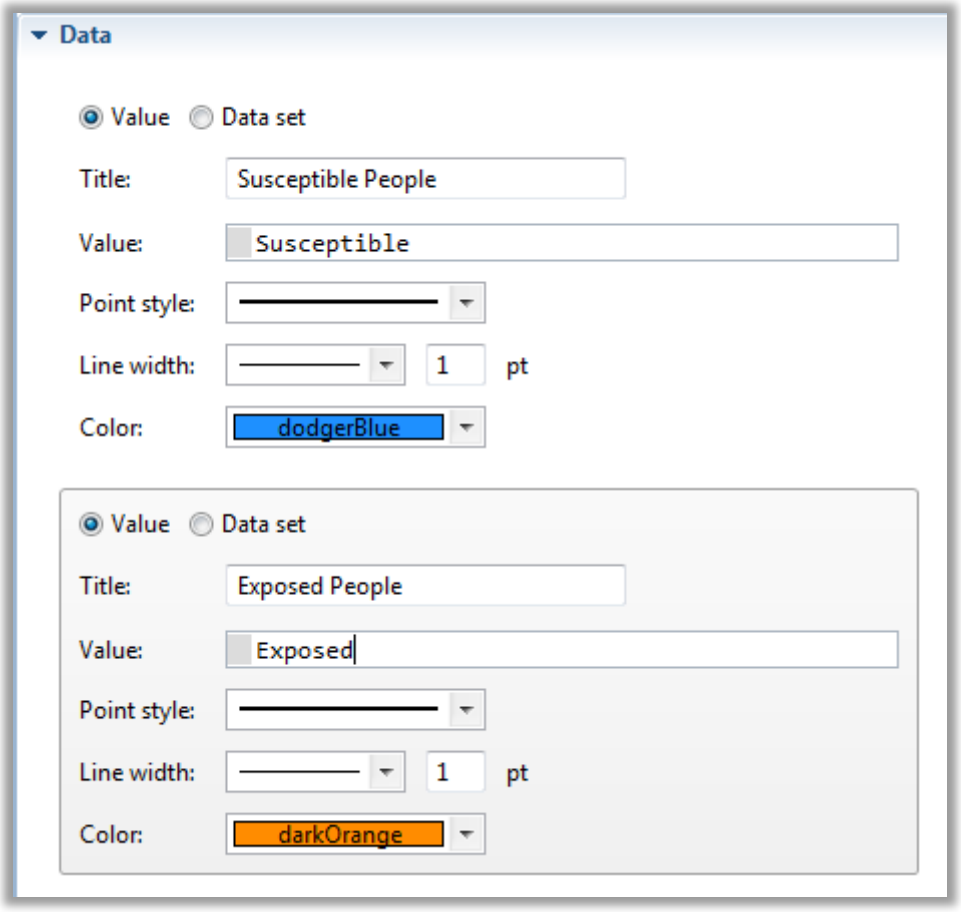

Le type de renforcement doit être R pour « Renforcement » et dépendant de la date (Clockwise).

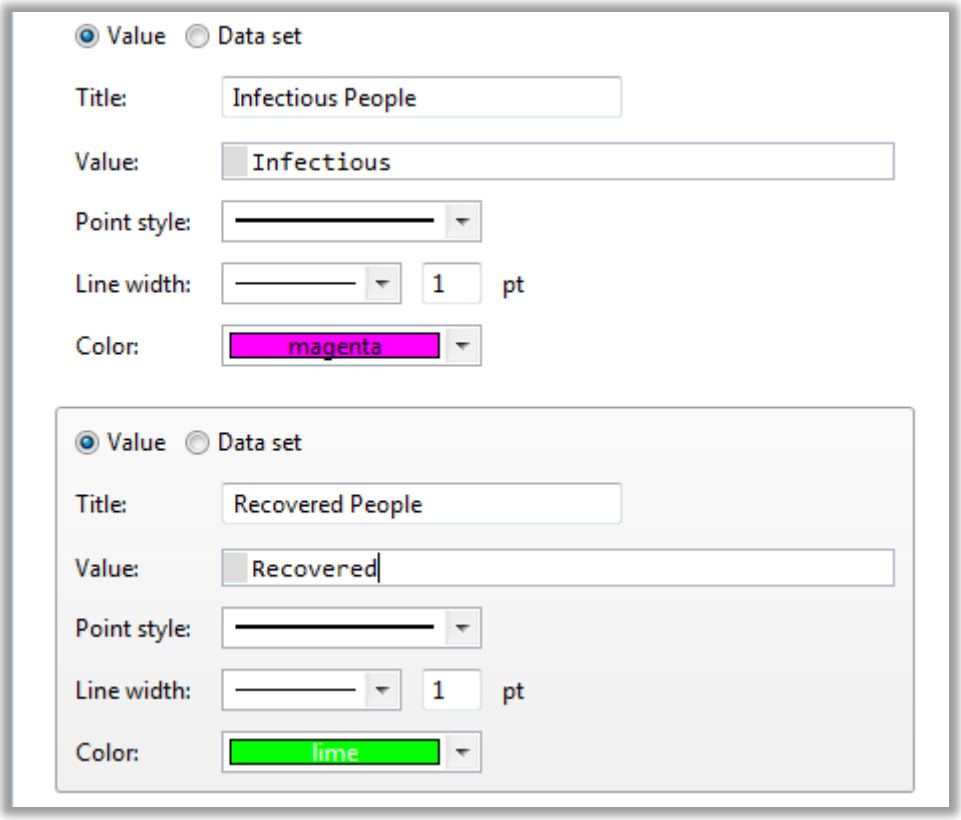


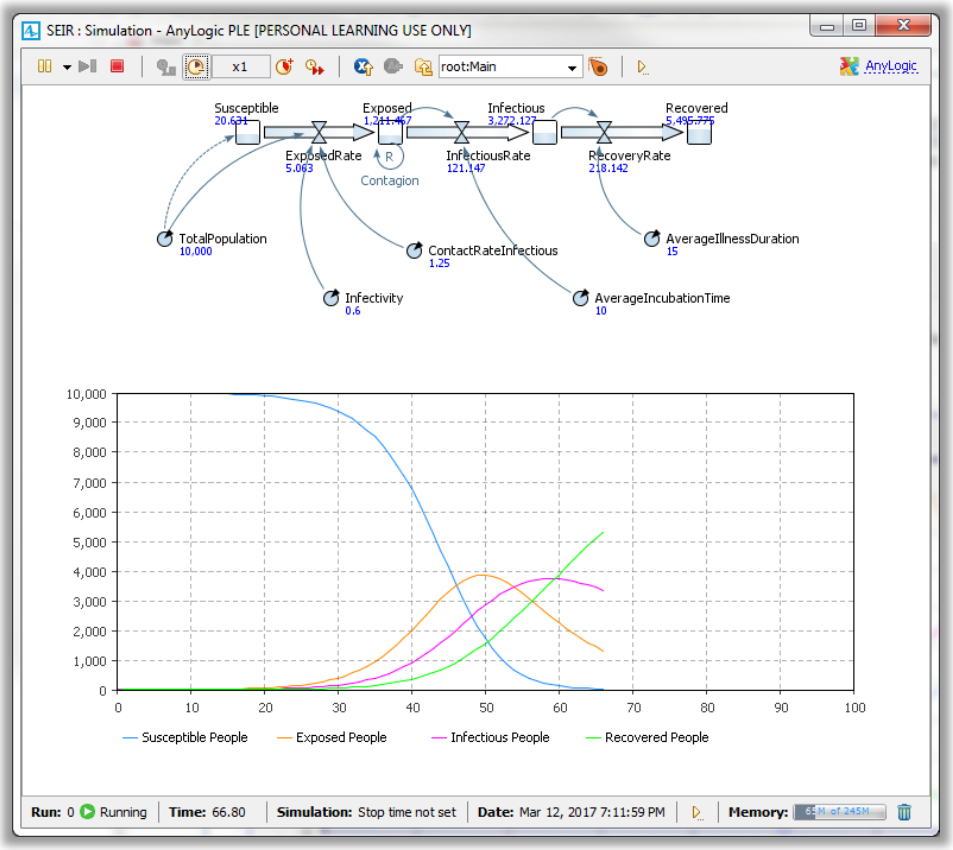

Il faut poser sur la feuille un élément graphique de type Time Plot.



Il faut ajouter 4 courbes.



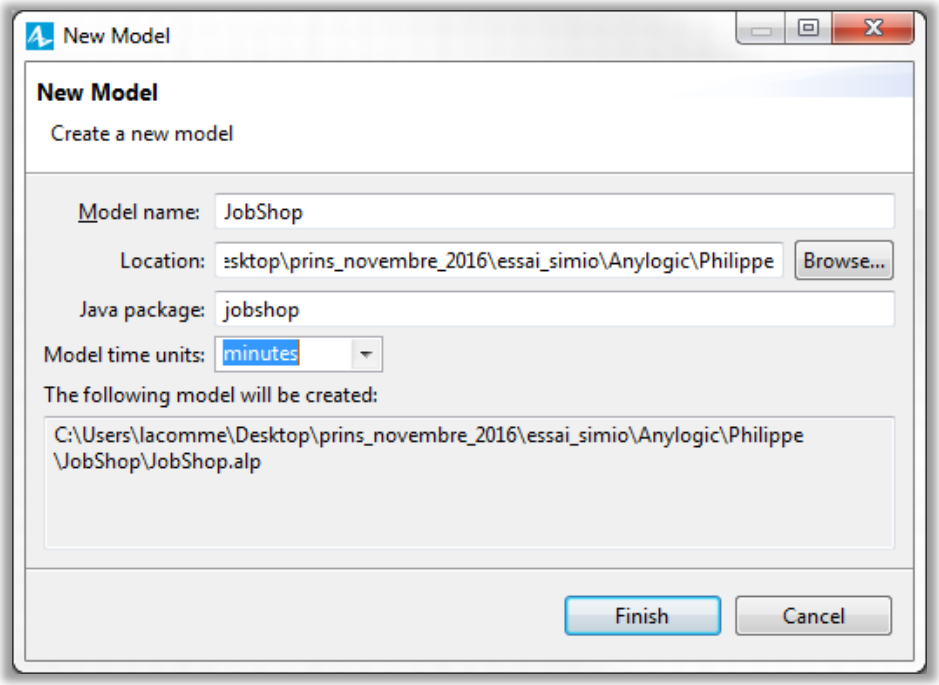






## 5) Modèle de simulation spatial et discrets

Le modèle est nommé JobShop pour respecter le choix réalisé dans le document pdf original mais le nom est un peu mal choisi.

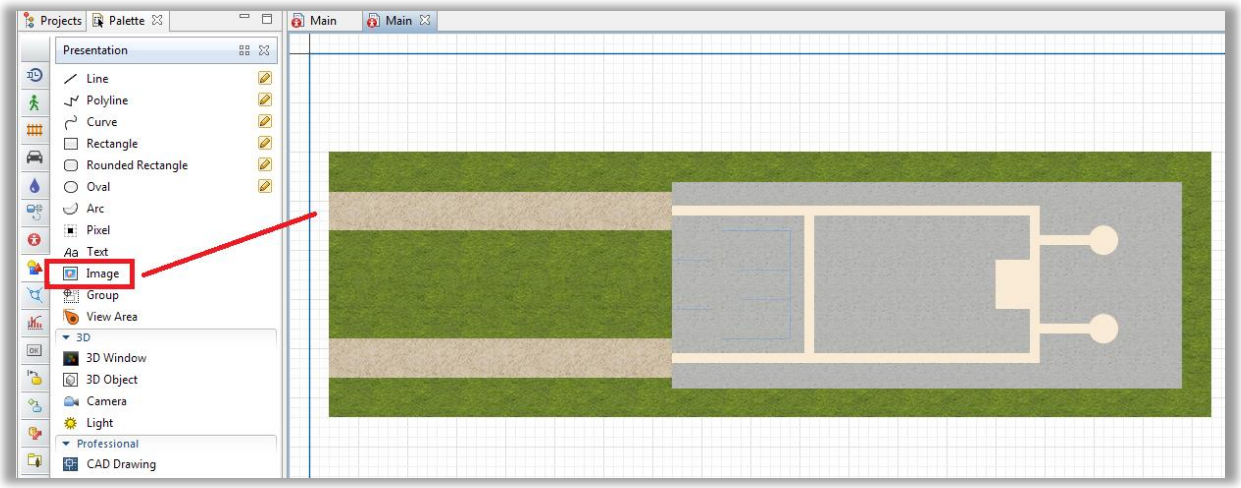
### 4.1. Création d'un modèle.



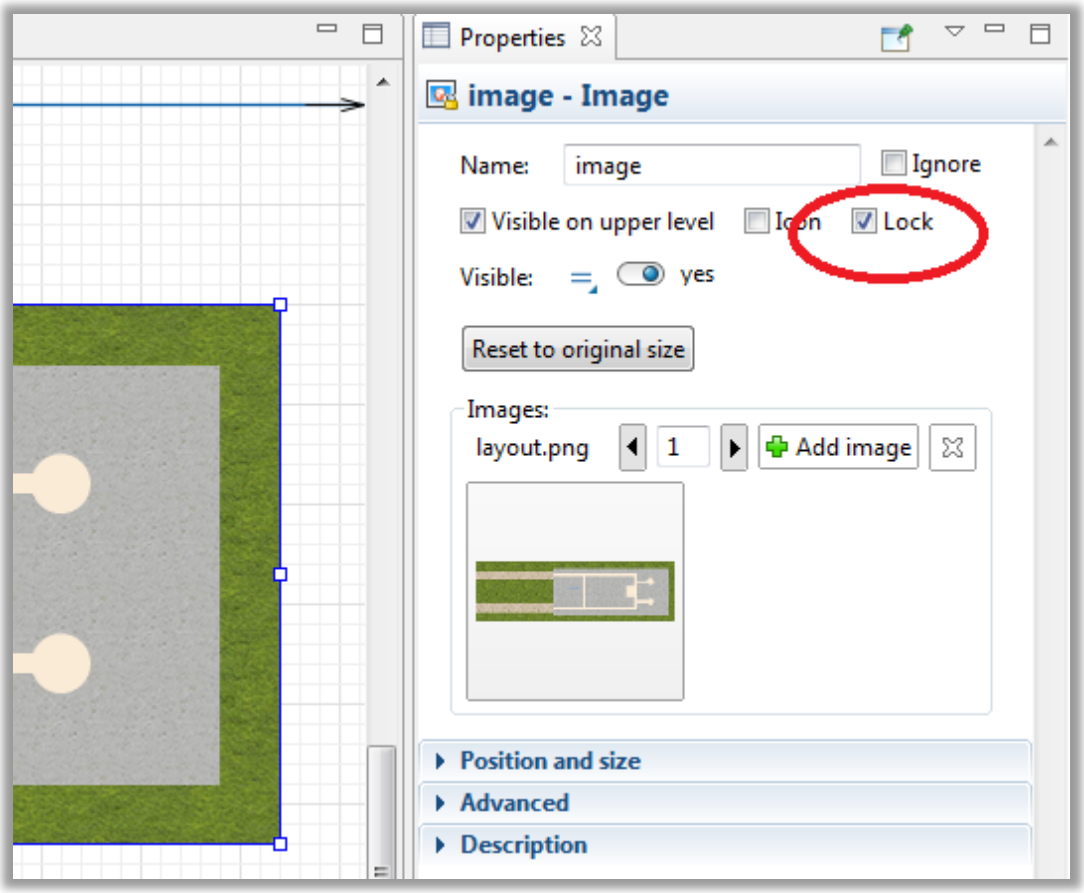

Il faut insérer une image et choisir l'image qui se trouve dans le répertoire suivant :

**C:\Program Files\AnyLogic 7 Personal Learning Edition\resources\AnyLogic in 3 days\Job Shop**

Cette image représente la vue aérienne des salles d'un atelier de production.

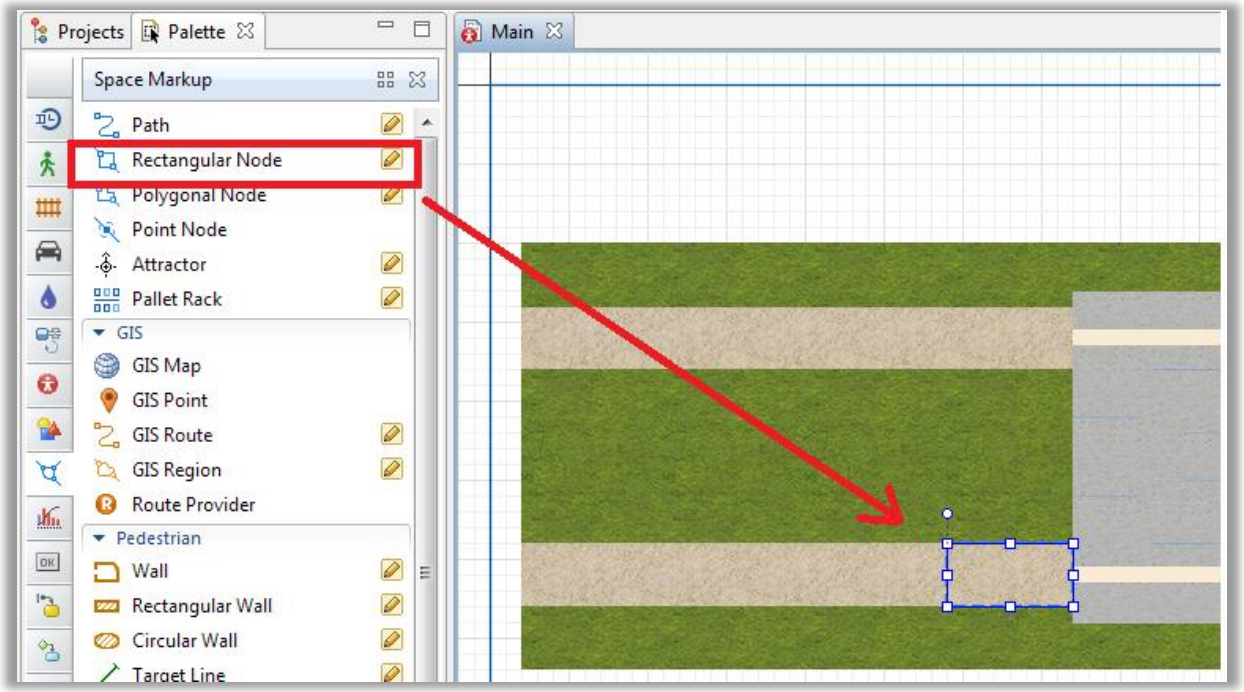



Une fois choisie la disposition de l'image, il est recommandé de passer l'attribut Lock à true, ce qui évitera des déplacements intempestifs de l'image.

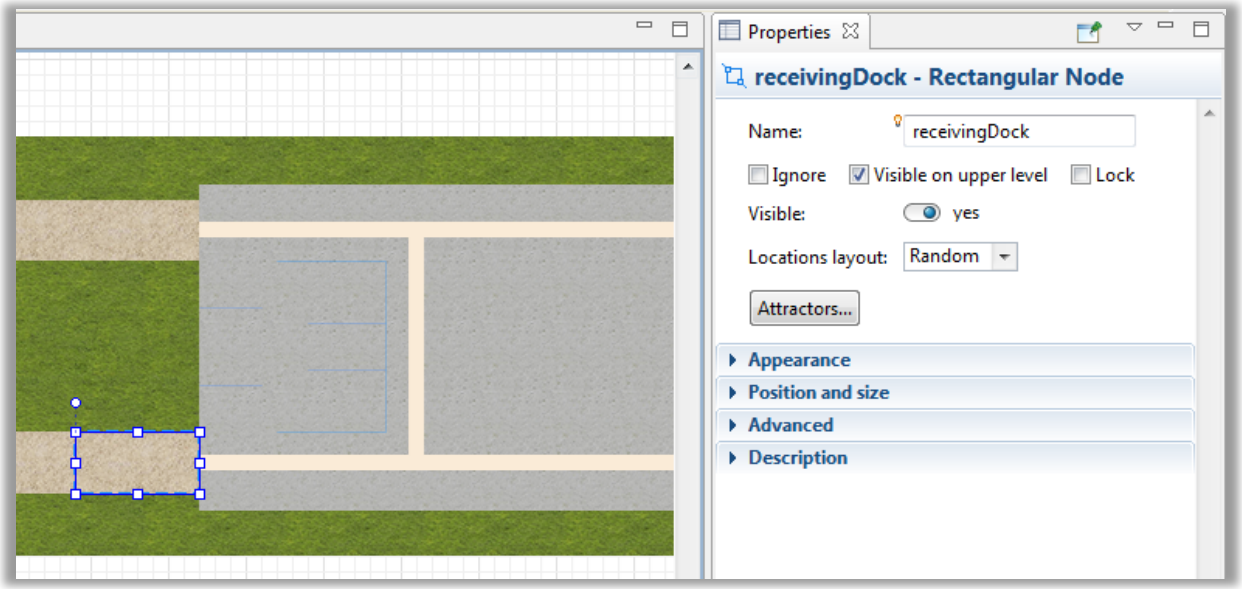


### 4.2. Création d'un réseau de transport

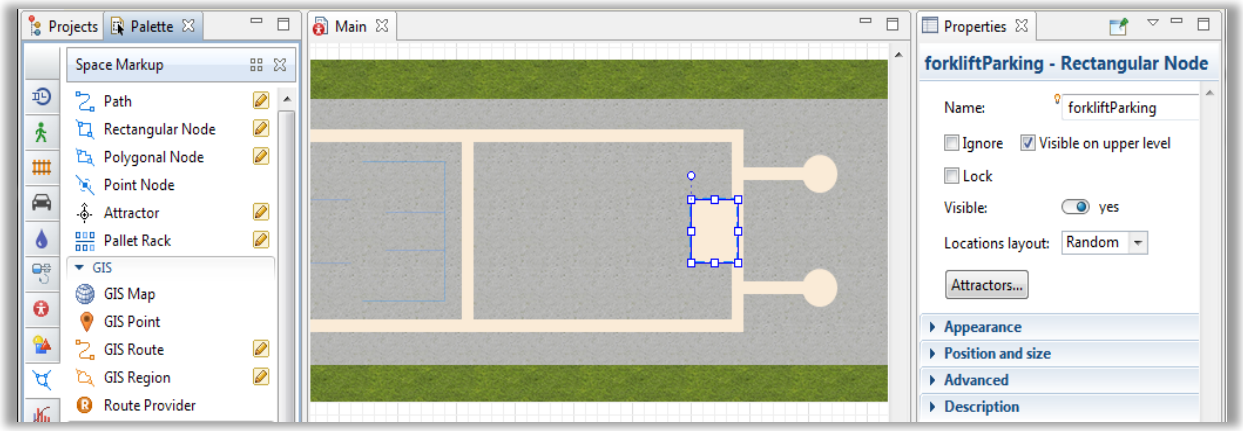
Un **Rectangular Node** (section **Space Markup**) est déposé sur le dessin.



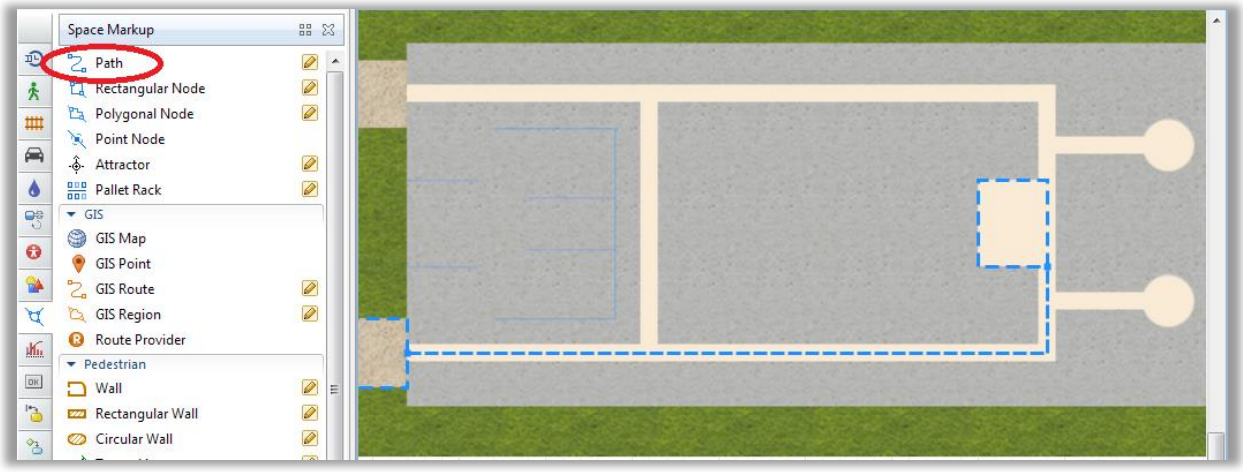

Ce nœud est nommé **receivingDock**.



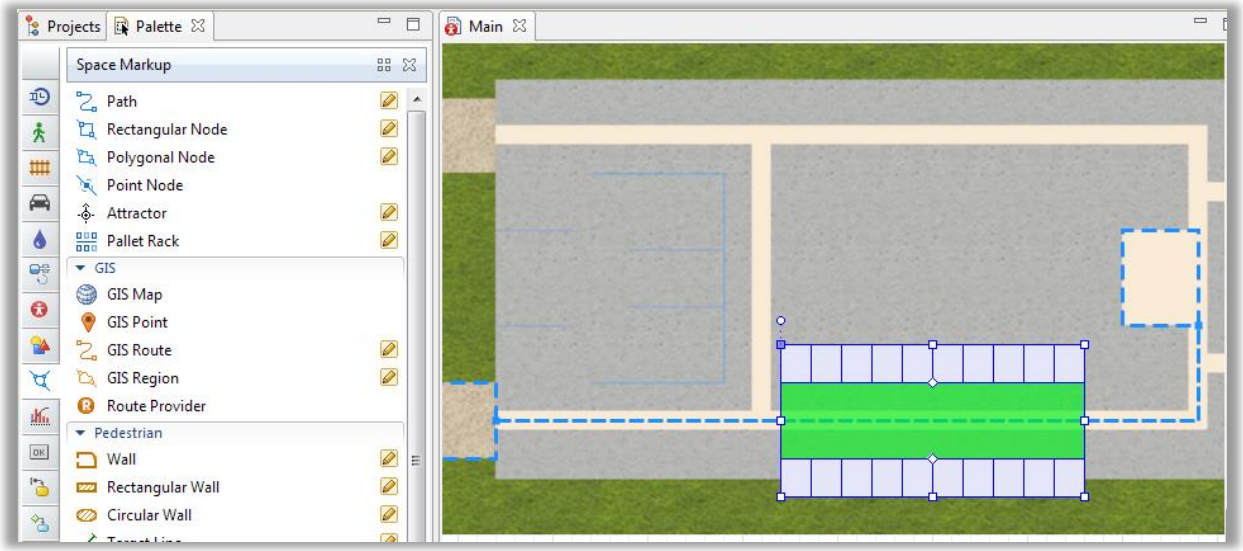
Un deuxième nœud est ajouté et nommé forkliftParking



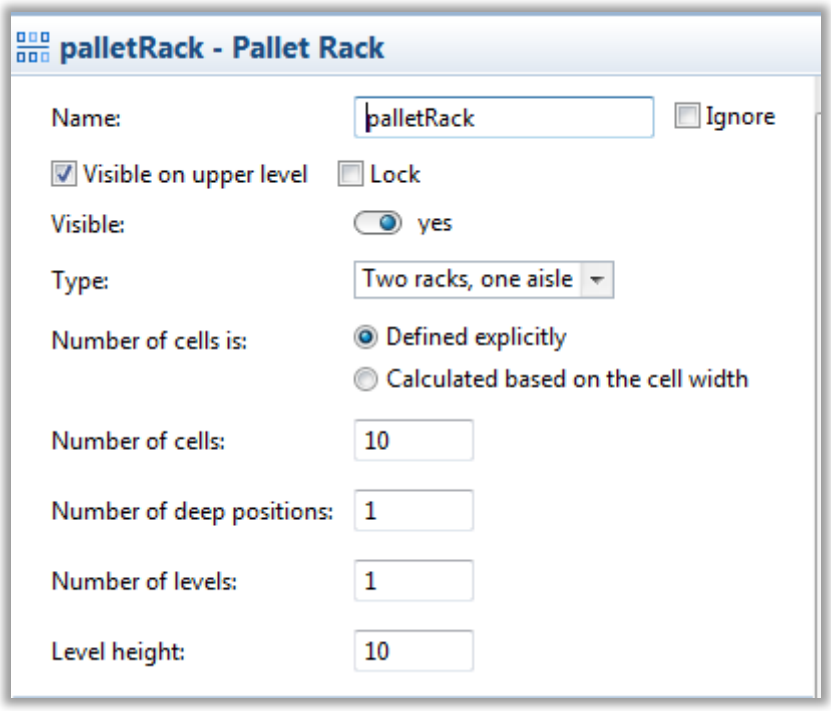

On définir ensuite un chemin Path entre ces nœuds, chemin qui servira à animer le déplacement des objets et des véhicules.



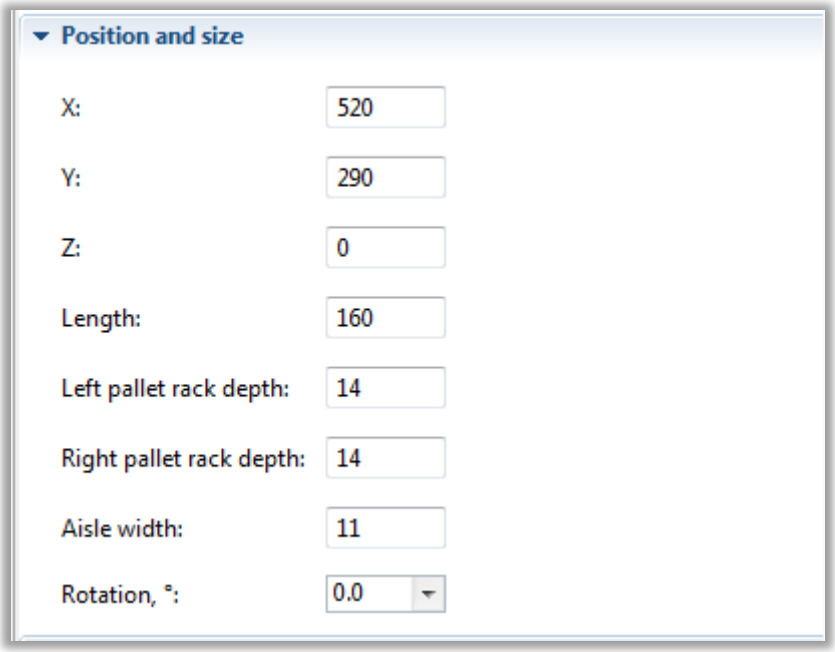

Il faut déposer une Pallet Rack sur le schéma en intersection avec le chemin. Attention, lors d'un positionnement correct (le rack est situé sur le chemin) une partie rectangulaire verte doit apparaître.



On définit une zone de stockage avec deux éléments et donc de type **Two racks and one aisles** configuré comme suit :

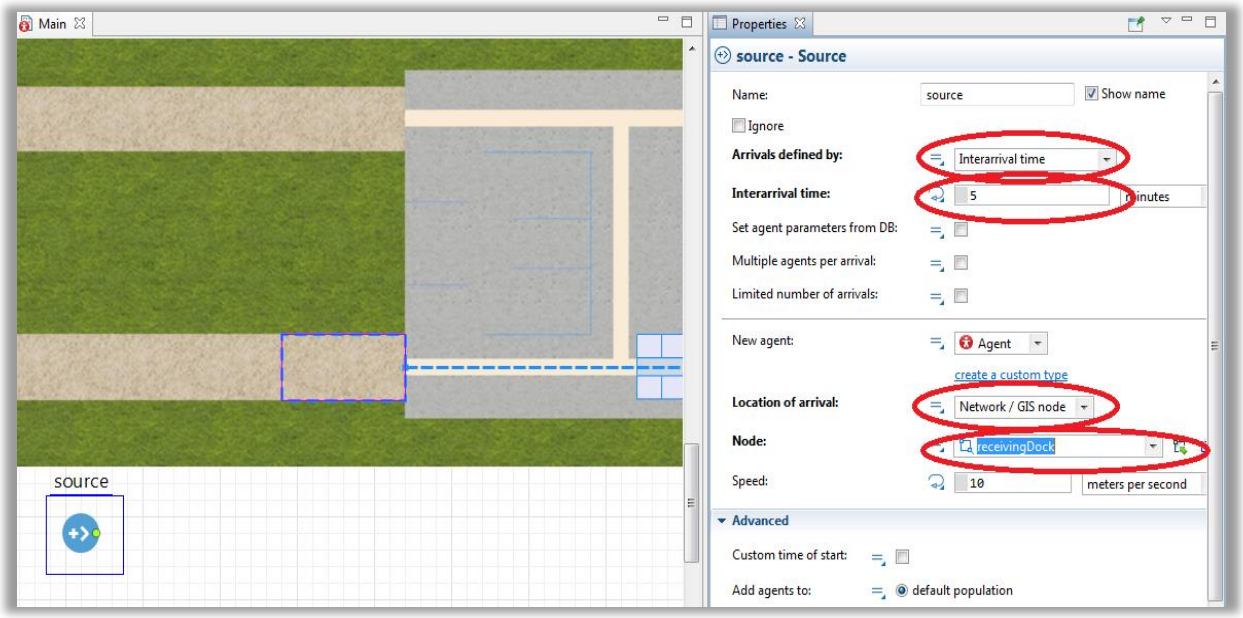



On peut ensuite modifier les paramètres de **Position and size** comme suit :

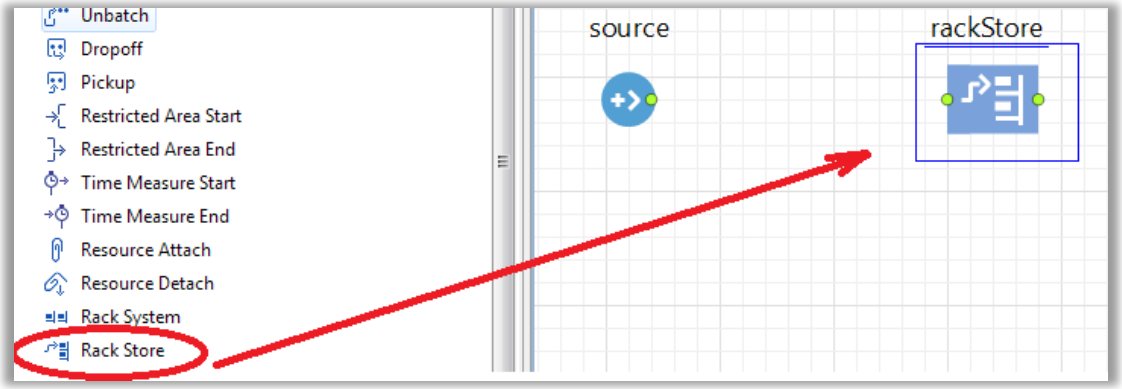


### 4.3. Définition d'un routage des pièces

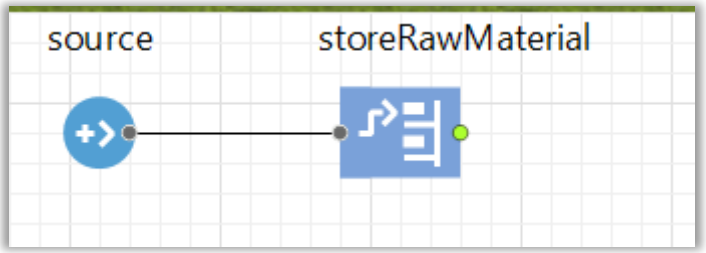
On utilise une source qu'on configure comme indiqué ci-dessous.



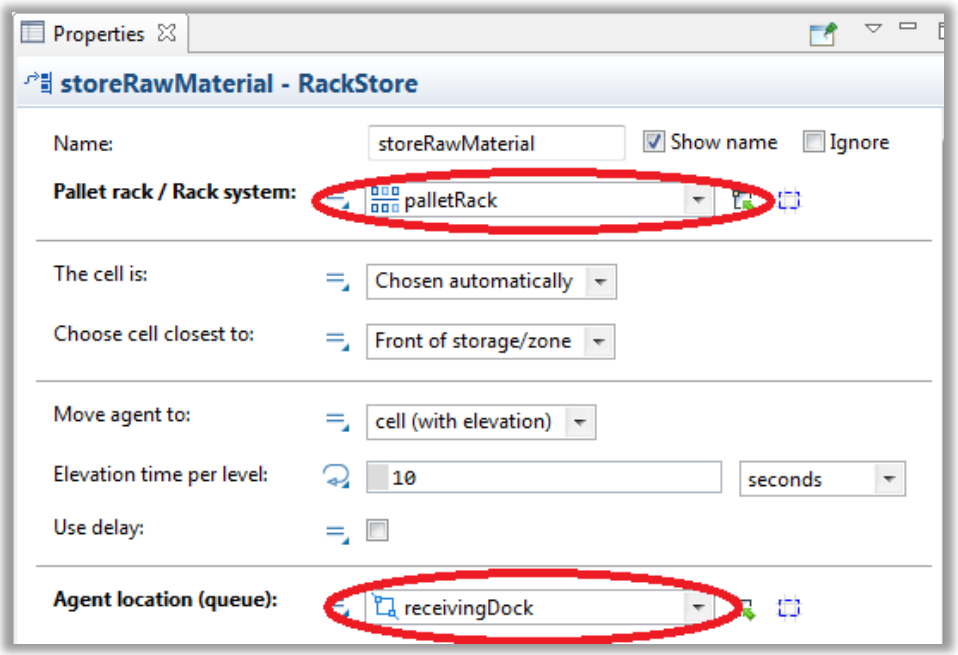

On ajout un **rackStore** à la suite.



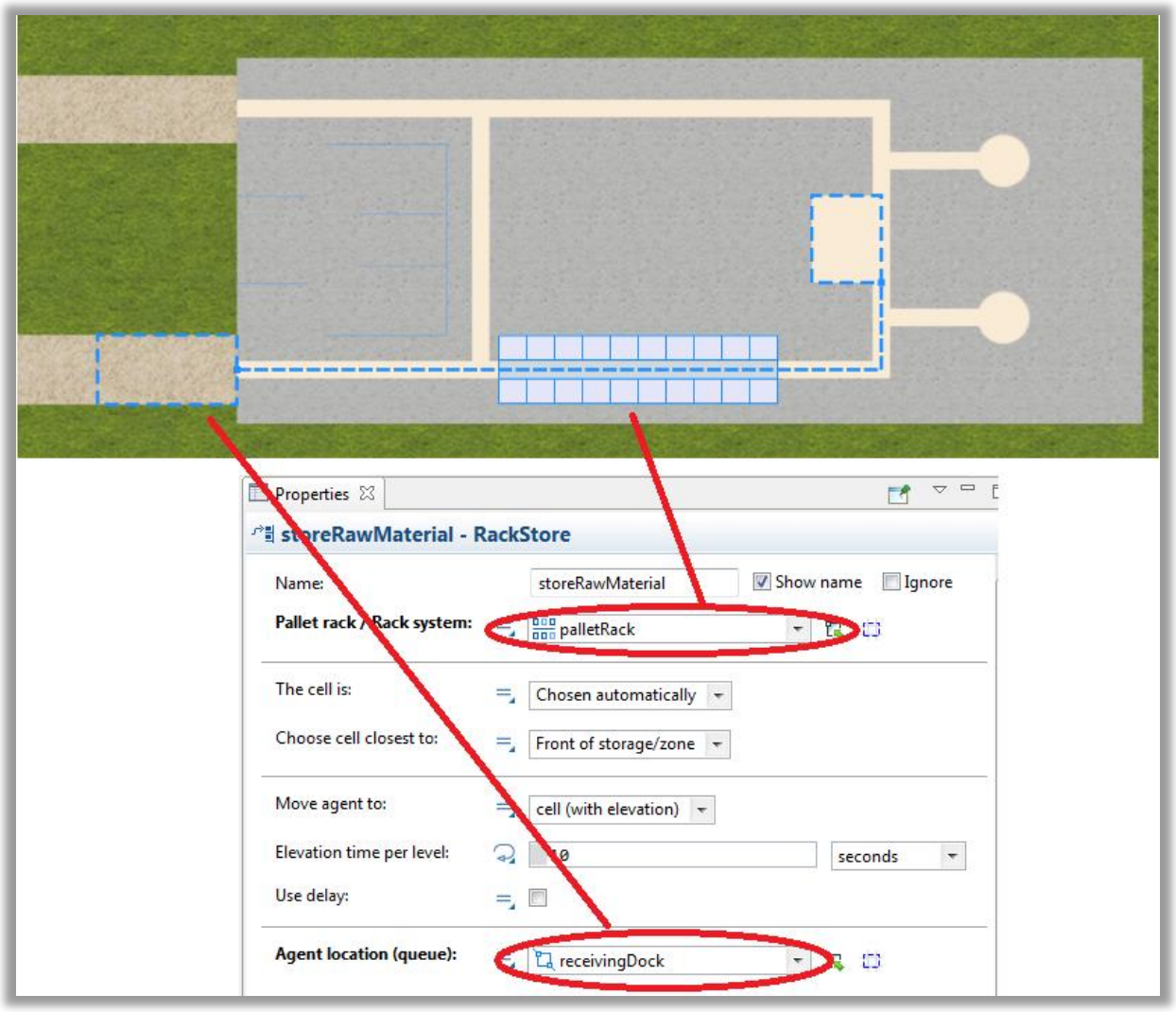

On le relie ensuite à la source.



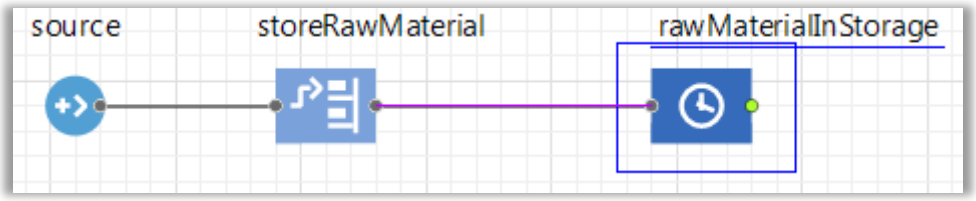
Il faut mettre à jour les attributs qui font le lien avec les objets graphiques…



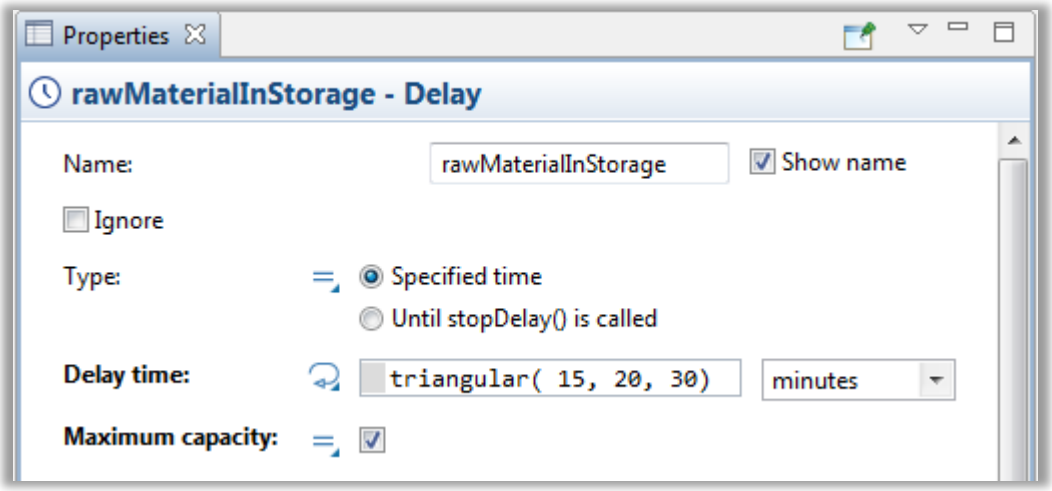

Les liens sont les suivants :



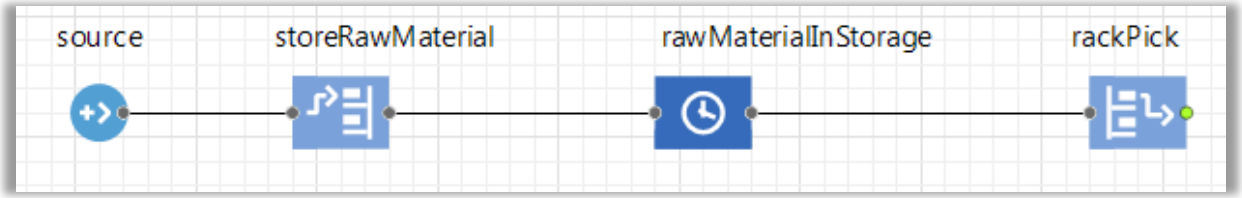
On ajoute ensuite un **délais**.



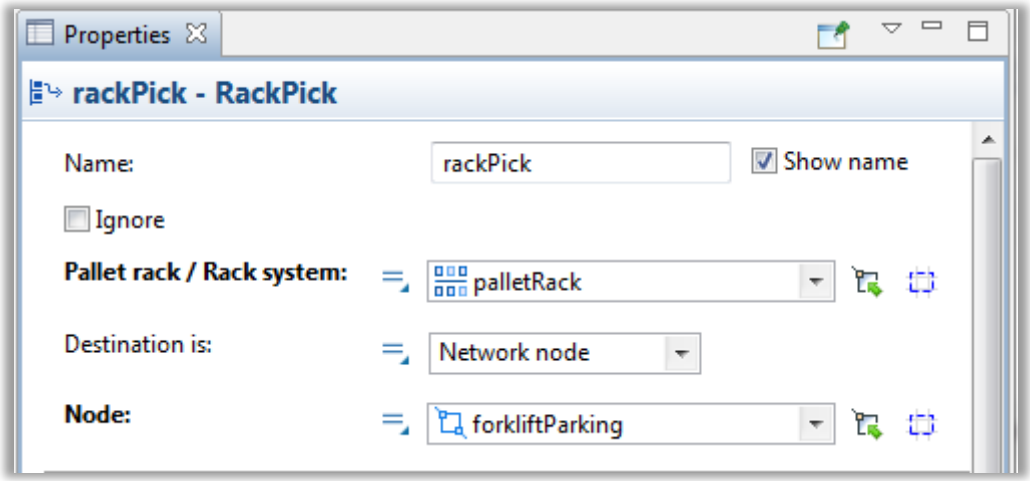

On le configure pour simuler une durée qui suit une loi triangulaire.



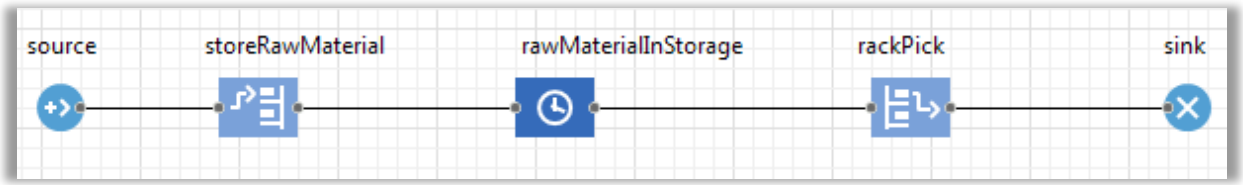

On ajoute un **rackPick**.



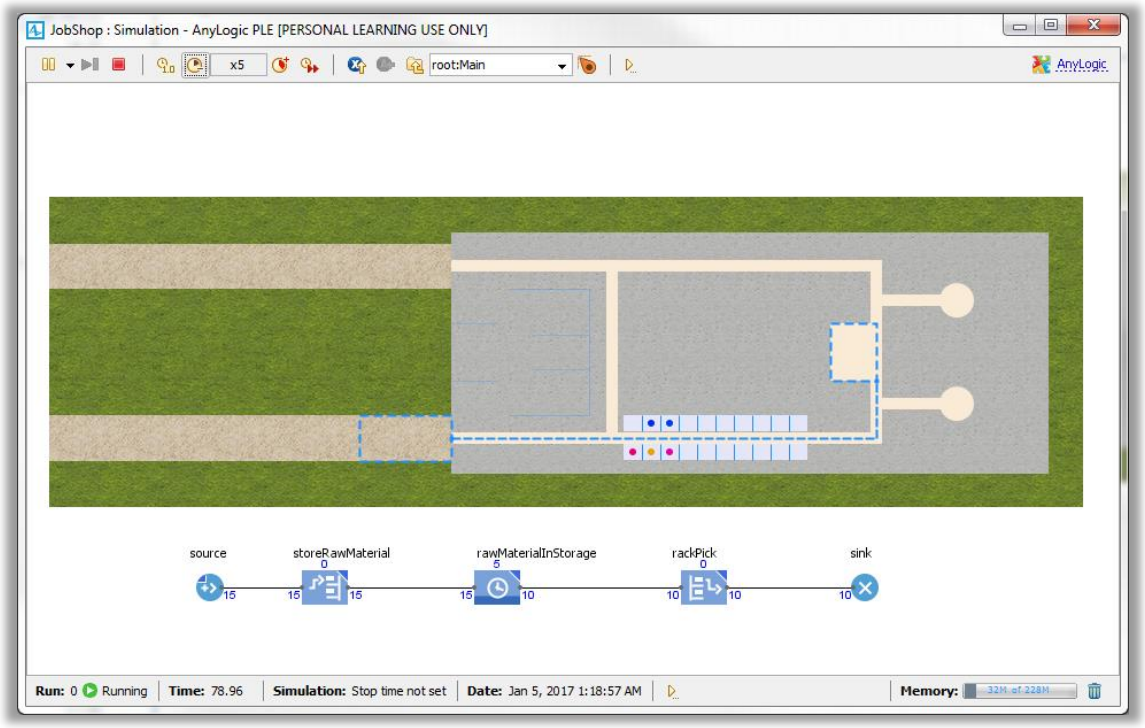

Et on relie cette opération aux différents objets dont le **palletRack**…



Il faut ajouter ensuite un puits.

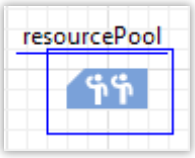



**Résultat d'exécution.**

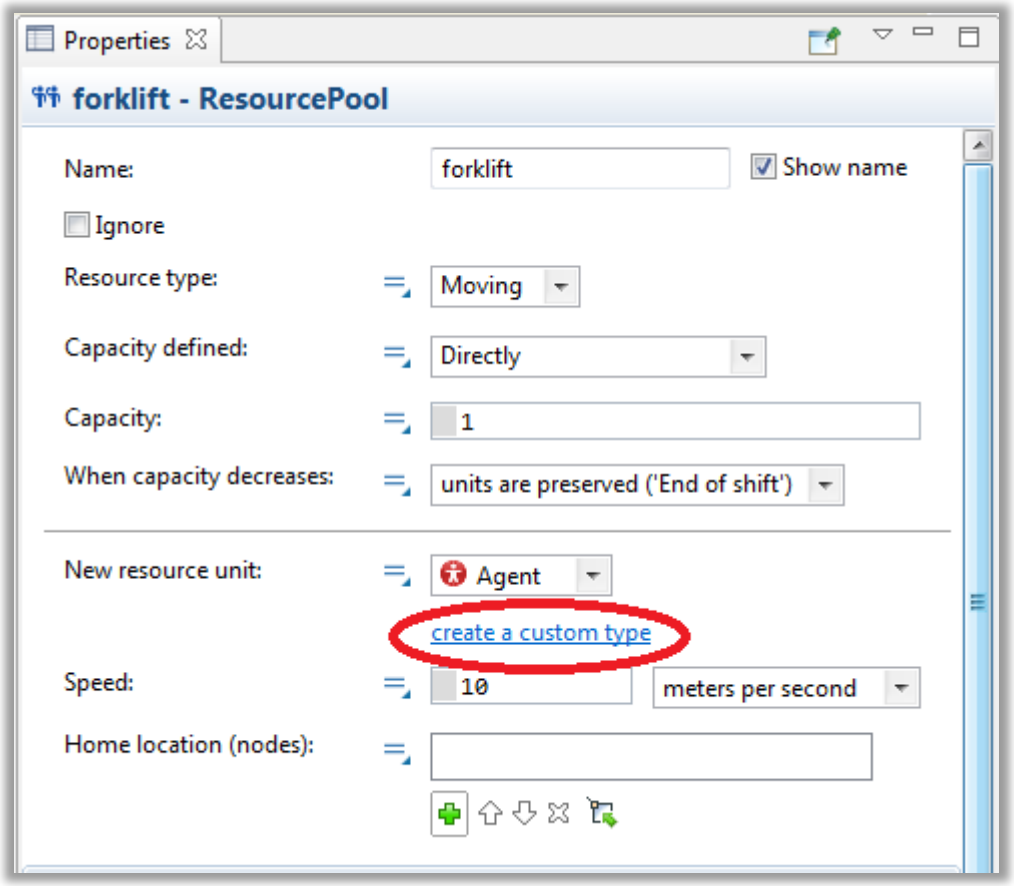



## 4.3. Ajout de ressources

Il faut ajouter un pool de ressources au modèle.

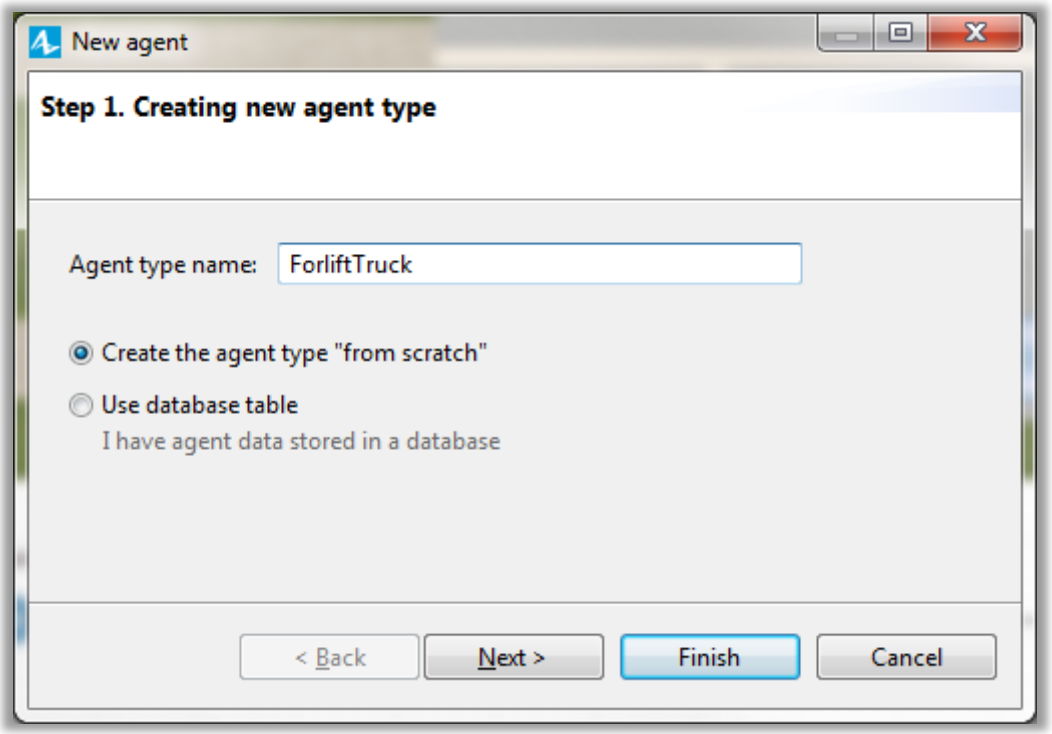



On nomme ce pool de ressources **forklift** en modifiant l'attribut **name**.

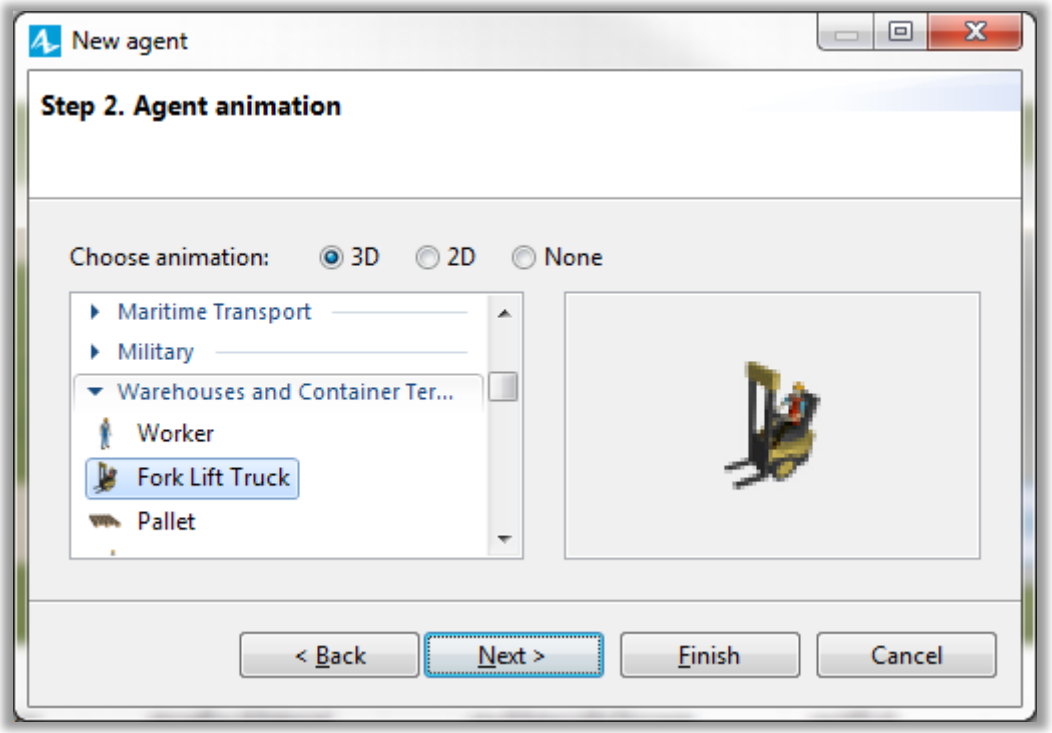



On va créer un type d'agent particulier pour modéliser les mouvements des chariots transportant les pièces.

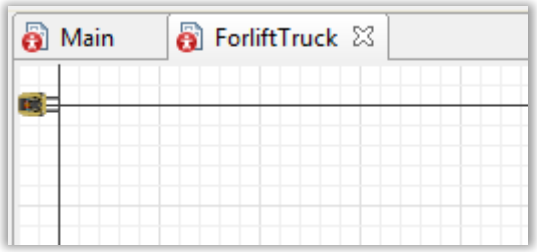
On nomme ces nouveaux agents : **ForkliftTruck**.



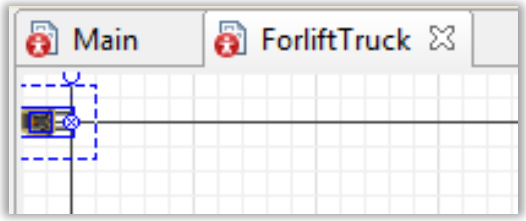
On choisit un objet dans la librairie d'objets 3D qui modélise au mieux un chariot.



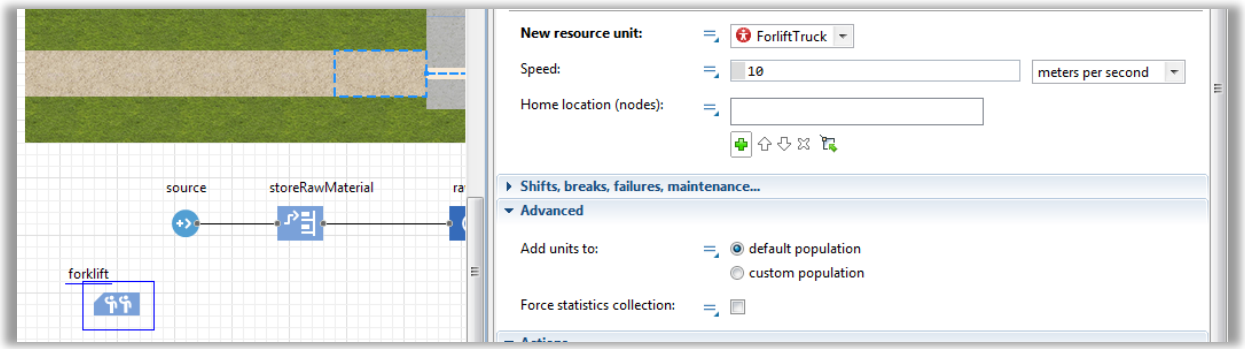

Une fois crée, le chariot apparaît dans un onglet nommé **ForkliftTruck**.



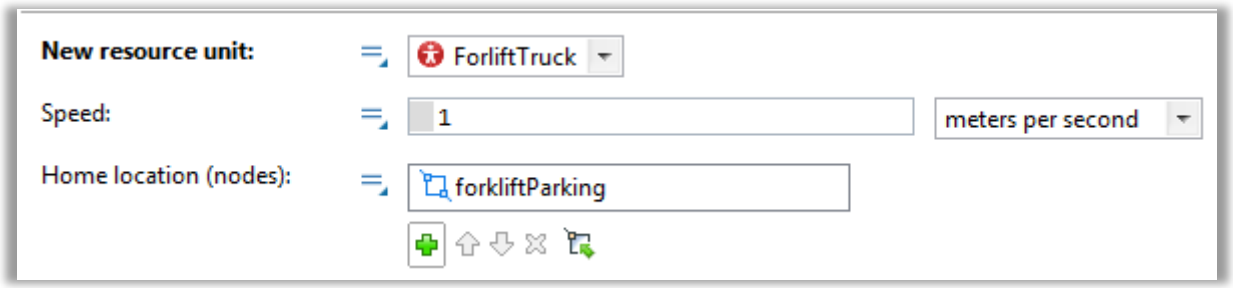

L'objet chariot peut être sélectionné à la souris et on peut accéder à ses propriétés.



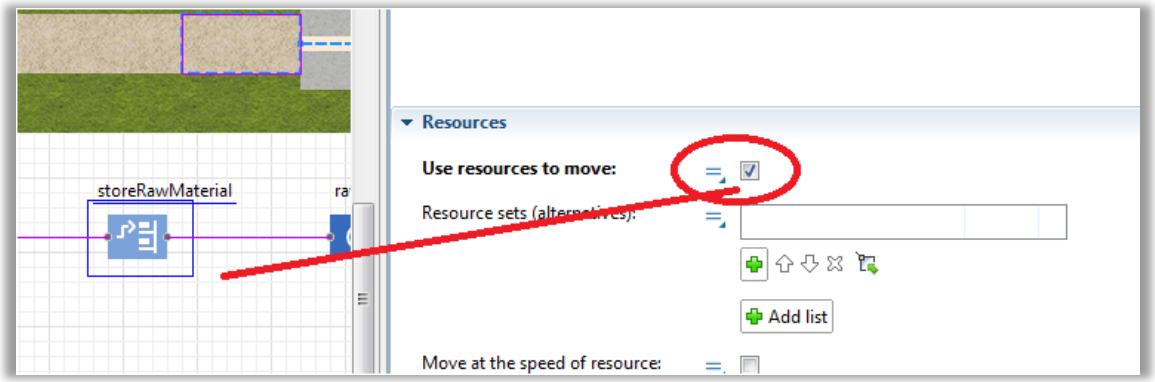

On revient au **forklif**



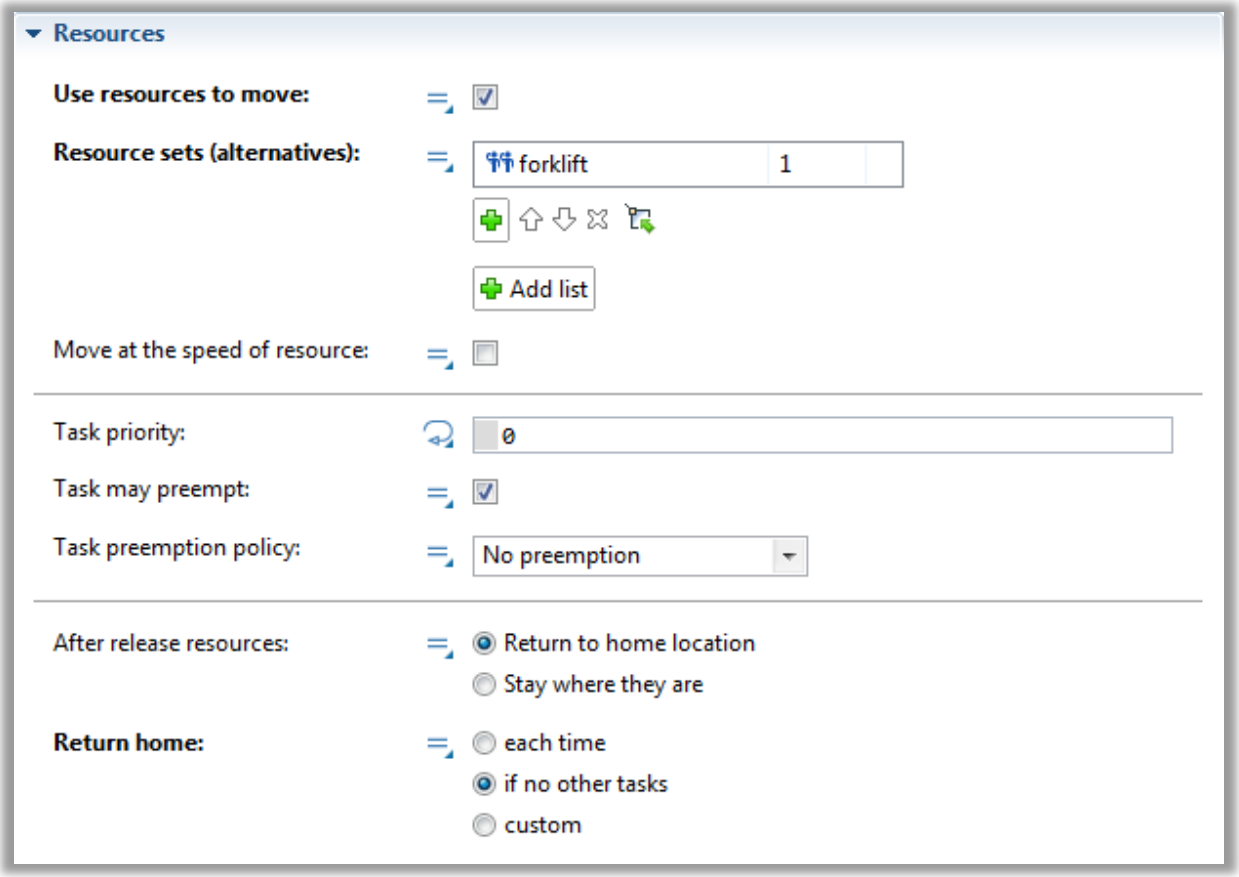
On spécifie alors la position initiale (dans le forkliftParking) et la vitesse en mètre par secondes.




On sélectionne le **storeRawMaterial** et on définit les ressources à déplacer en cochant la case dans l'onglet **Resources**.

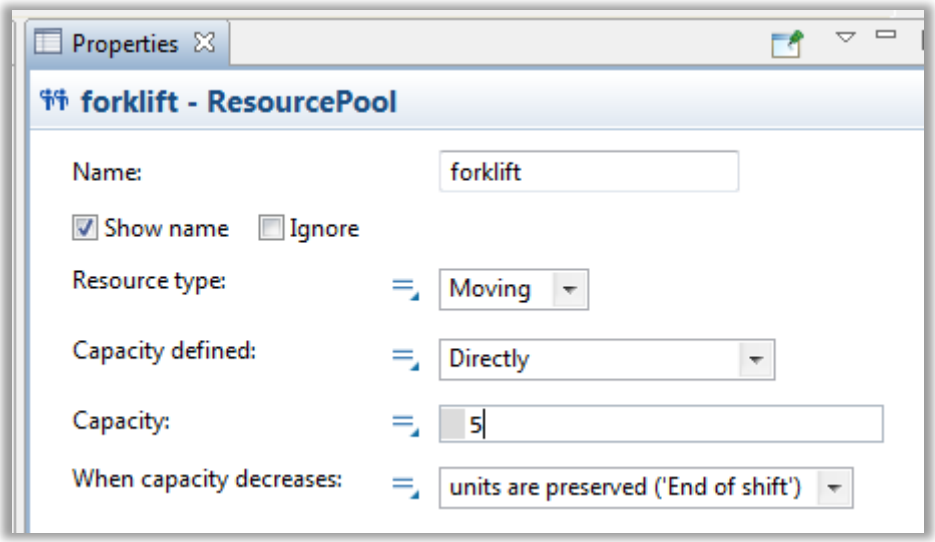



On définit ensuite les règles gérant les déplacements des chariots.

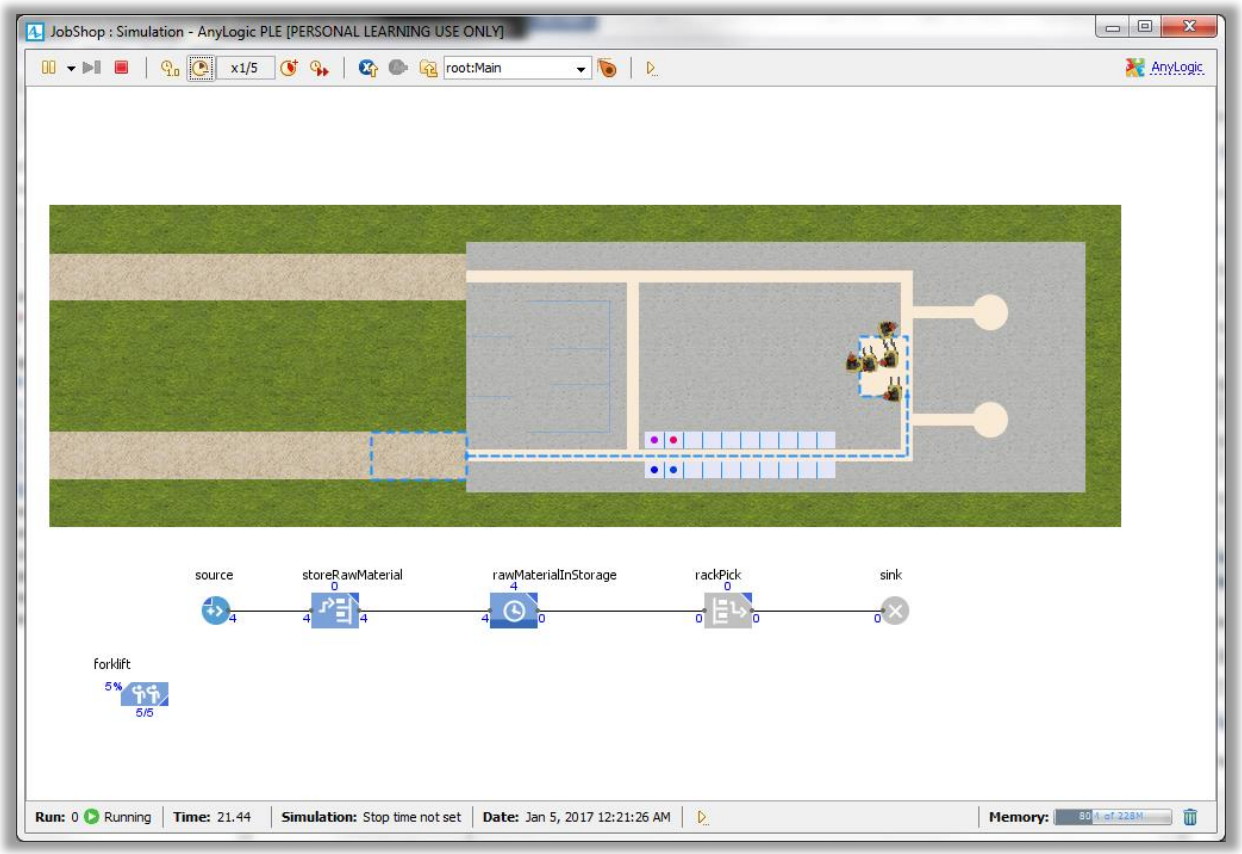


### 4.5. Exécution du modèle

Afin d'obtenir un modèle intéressant on peut fixer à 5 le nombre de chariots.

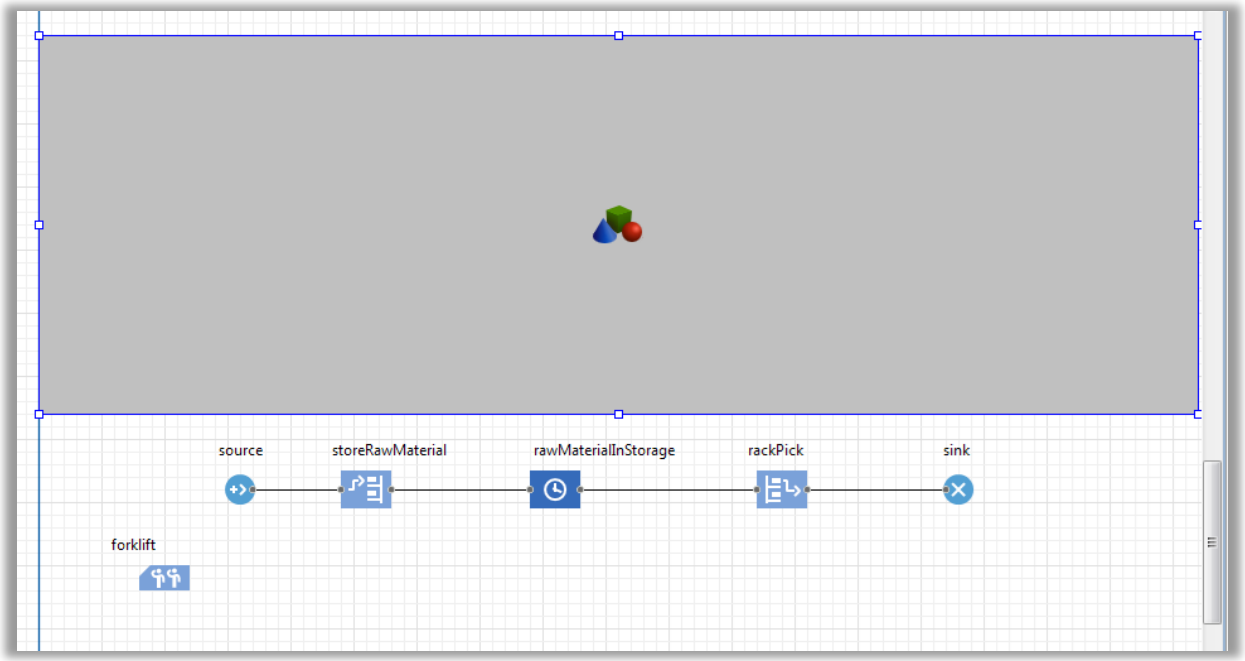



Le résultat est celui fournit ci-dessous :

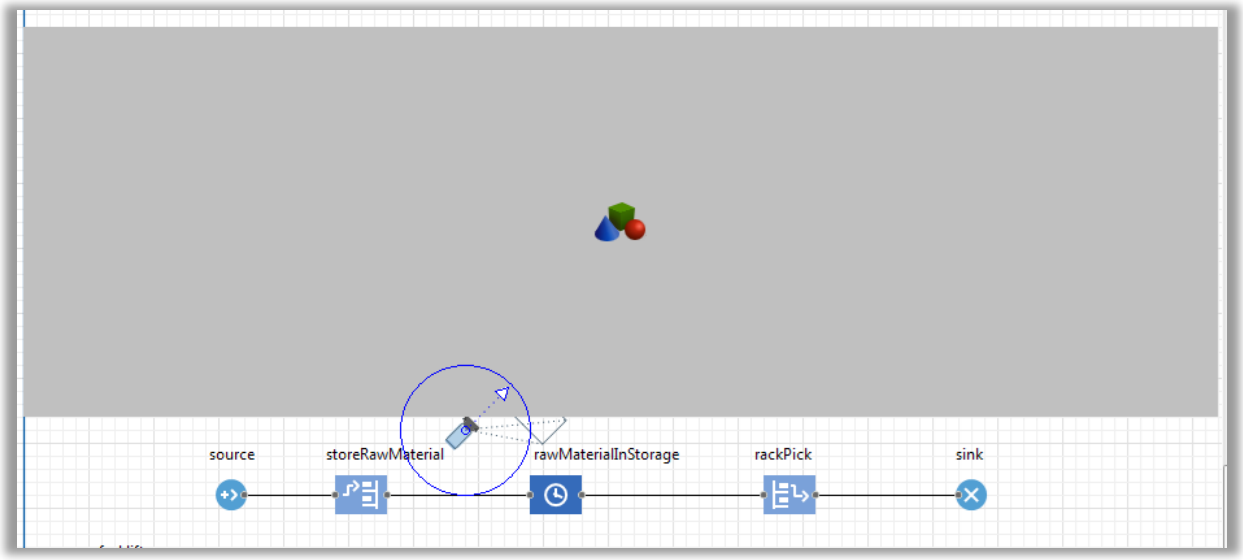


### 4.6. Ajout d'une animation 3D

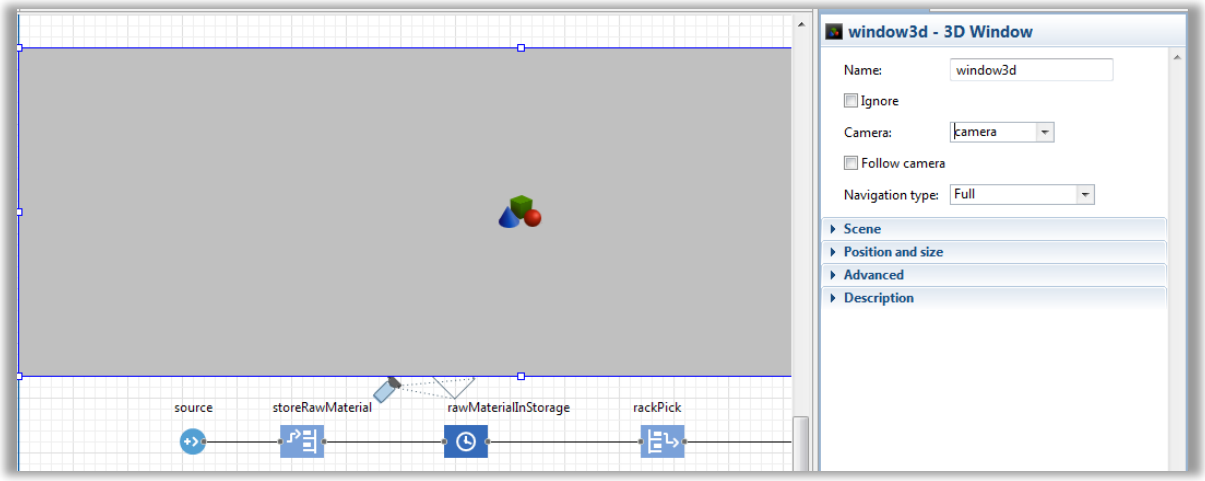
Il faut poser un objet 3D Window sur le modèle de sorte qu'il se présente comme ceci :




Il faut ensuite poser et orienter une caméra.



On fait le lien entre la scène et la caméra : on modifier les attributs de windows3d comme suit :




On obtient à l'exécution une vision 3D de la scène.