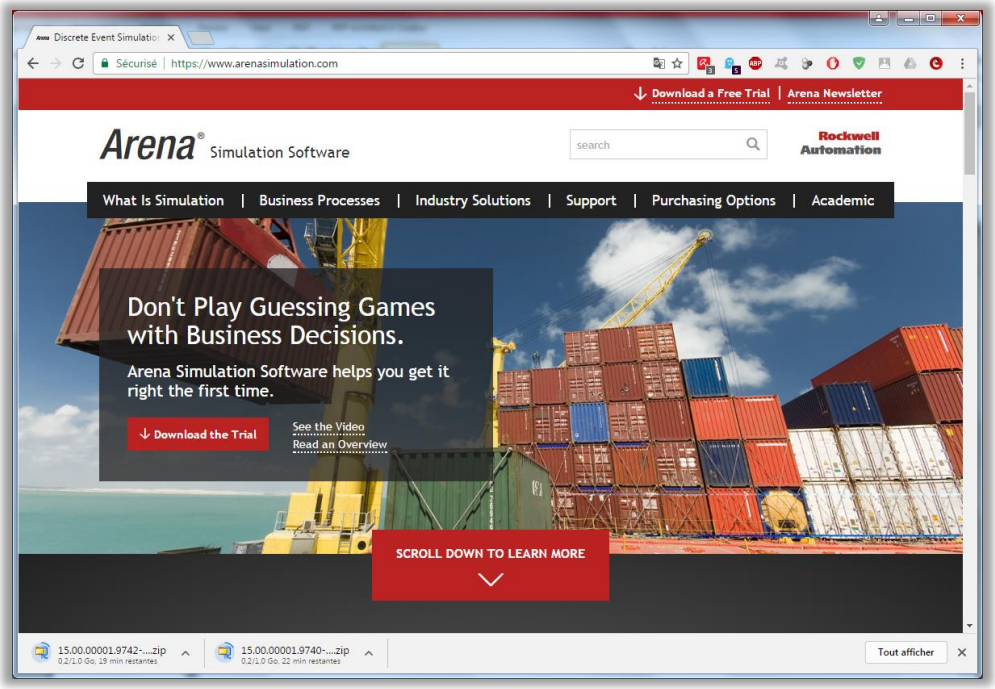
Utilisation d'ARENA

Auteurs : P. Lacomme (placomme@isima.fr)
D. Lamy (lamy@isima.fr)

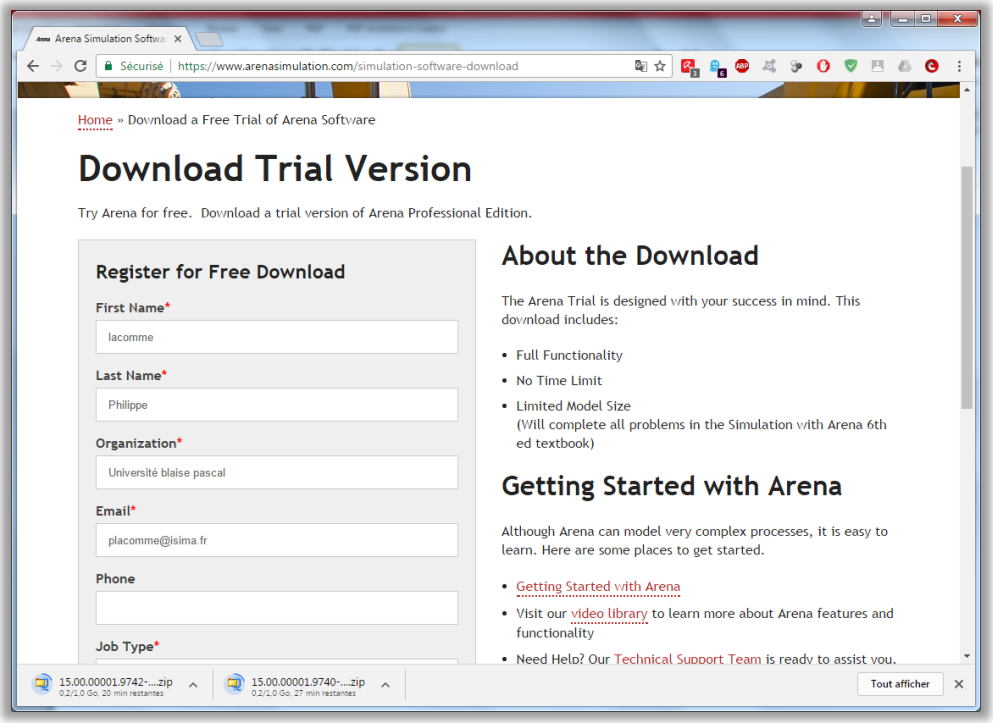
Date de création : Janvier 2017

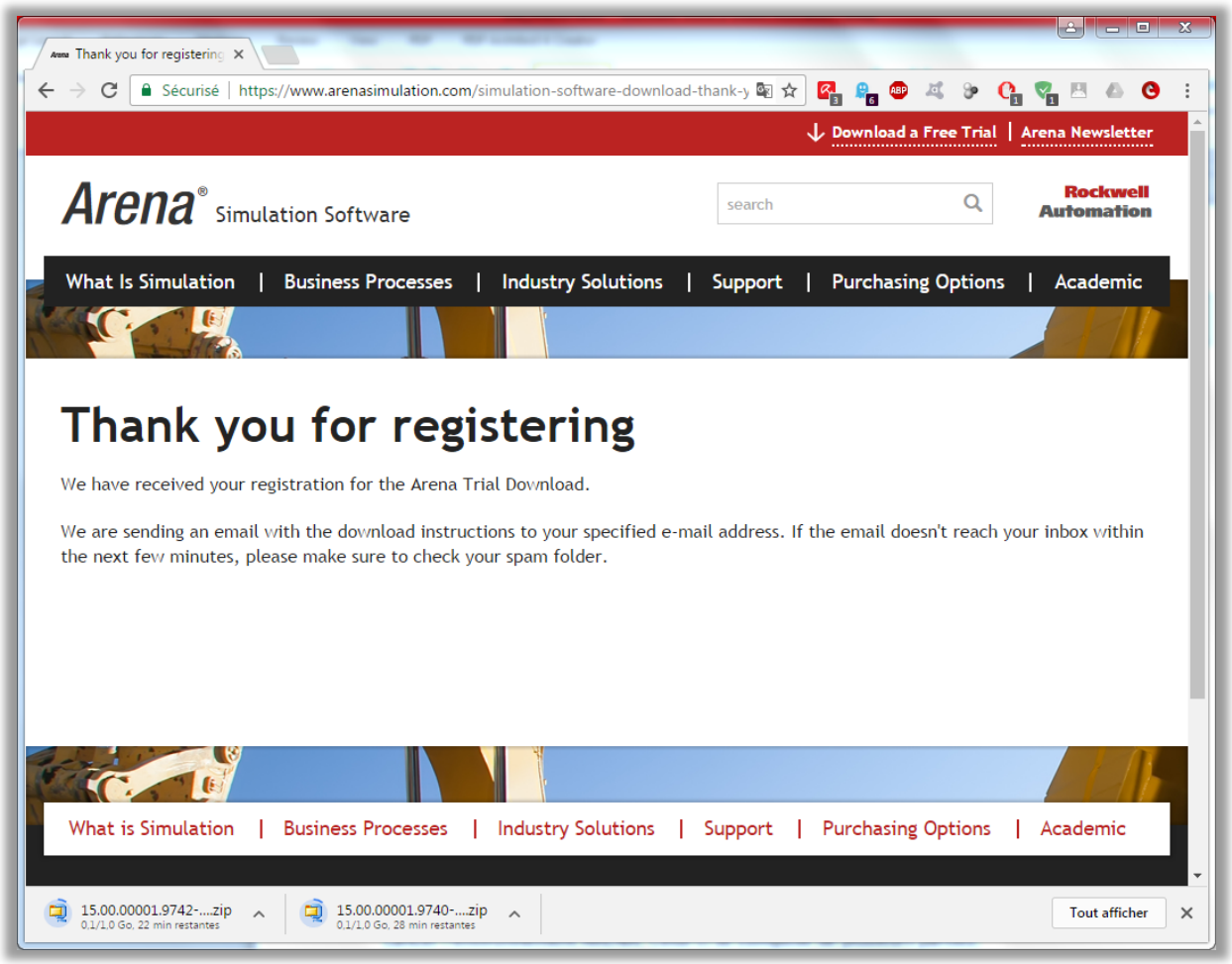
1) Installation

Le logiciel de simulation est disponible à l'adresse suivante : https://www.arenasimulation.com/



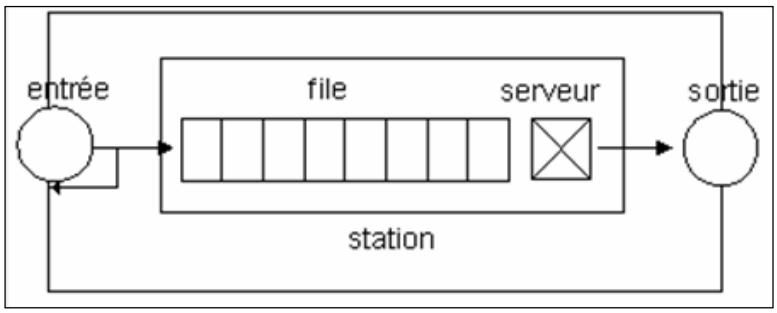
L'obtention du logiciel nécessite de remplir un formulaire et attendre la réception d'un mail.





2) Simuler une file MM1

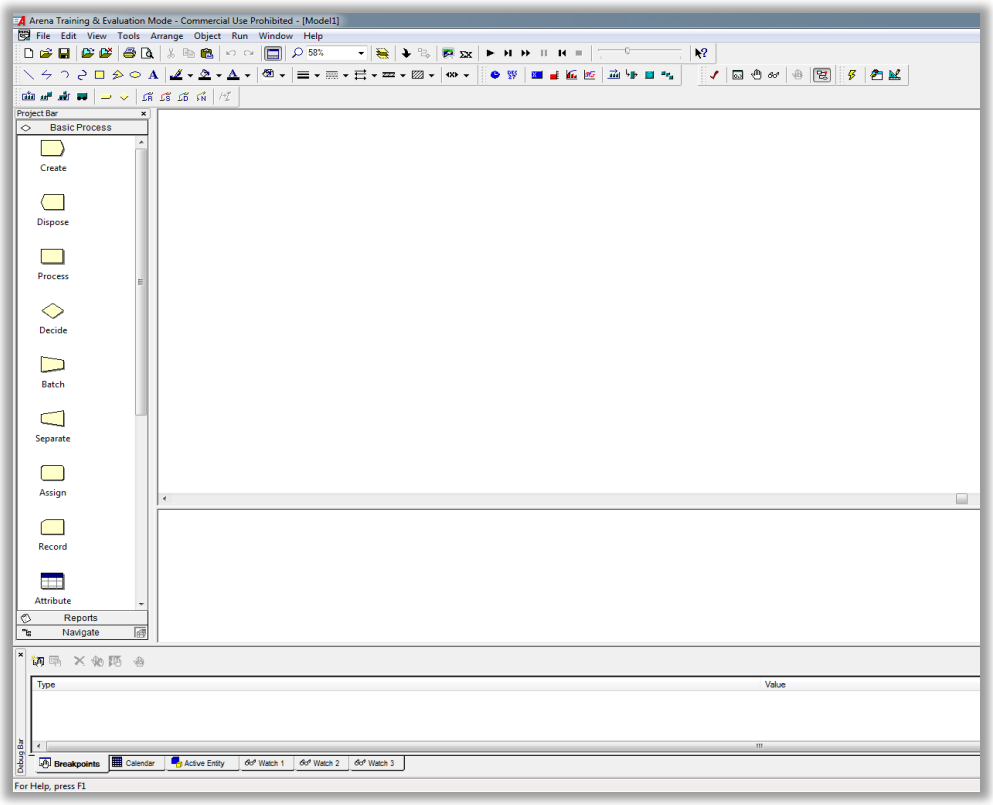
Le système à simuler se compose d'une source (entrée), d'une station (serveur + file d'attente) et d'une sortie.



2.1. Buffers de capacité infinie

Lancer l'environnement ARENA. Celui-ci se compose de plusieurs parties:

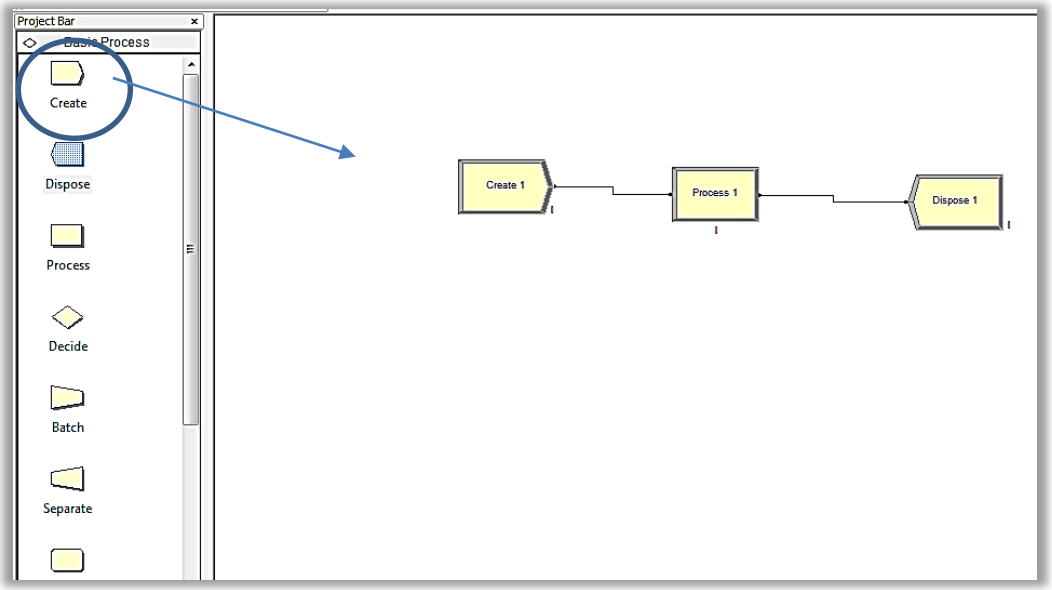
- Une librairie avec des objets prédéfinis dans la partie gauche;
- Une zone au centre pour définir le modèle;
- Différents onglets dont un en particulier intitulé Run.



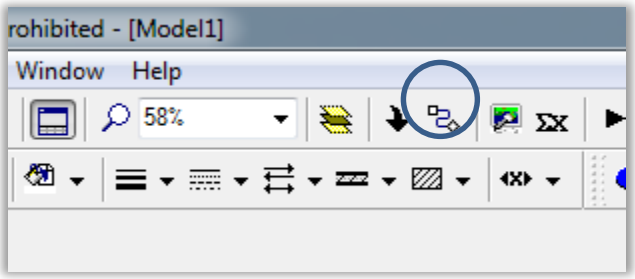
<u>Il faut placer sur la grille :</u>

- Une source (**Create**)
- Un server (**Process**)
- Un puits (**Dispose**)

Ces objets se situent dans l'onglet Basic Process dans le volet gauche.

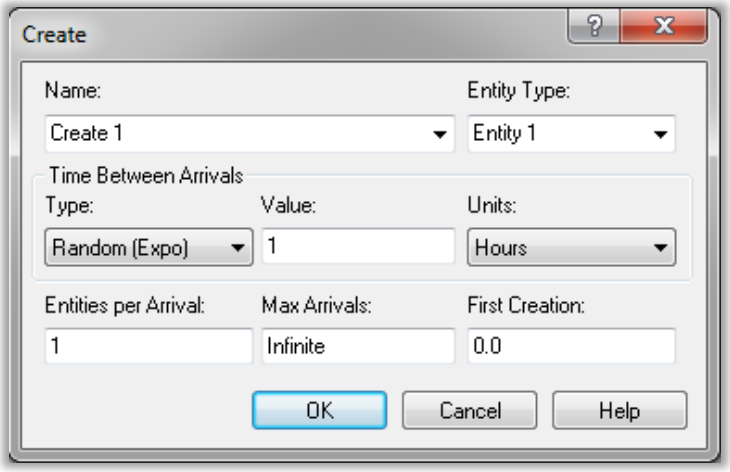


Normalement ces blocks sont automatiquement connectés. Dans le cas où ils ne le seraient pas, il faut joindre un bloc à un autre en utilisant l'icône 'connect':



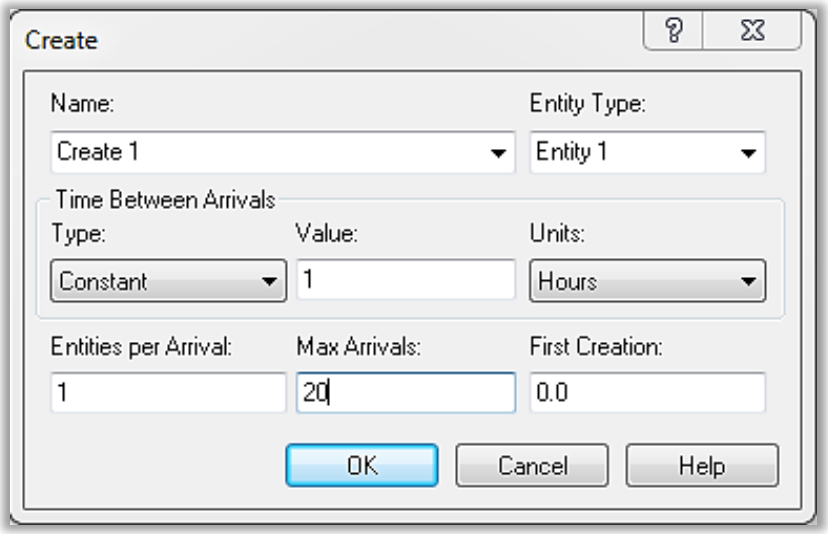
Paramétrage de la source

Il faut rentrer dans le menu de paramétrage de la source en double-cliquant sur elle.



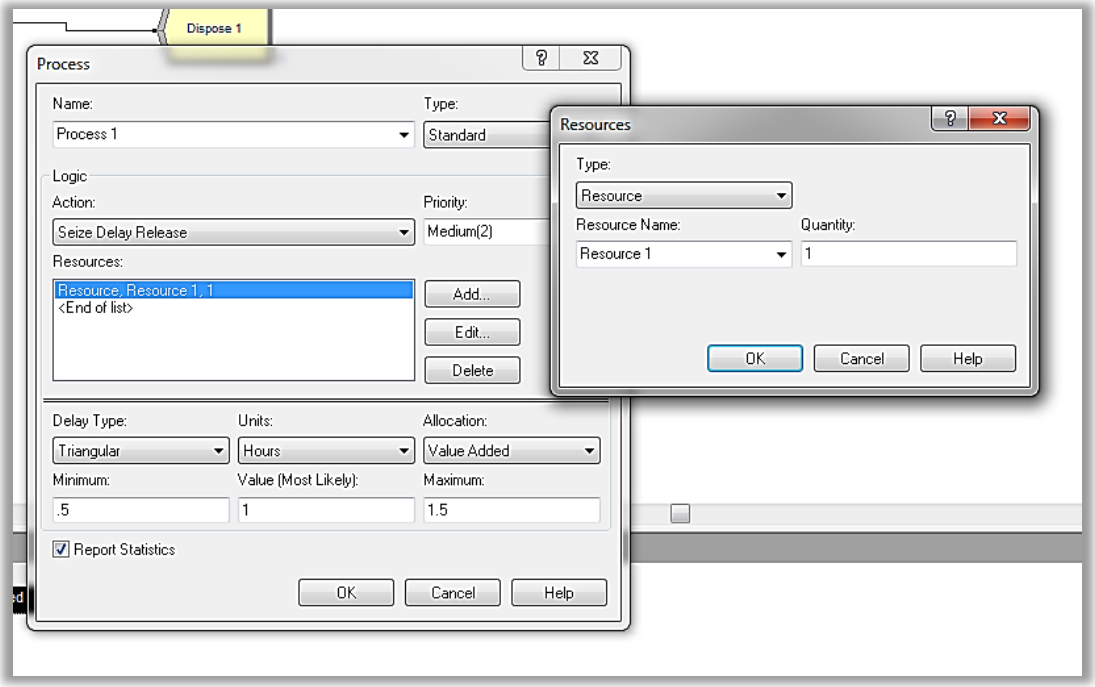
Par défaut, une loi exponentielle de paramètre 1 est proposée. Dans le menu déroulant, une loi constante est également disponible, ainsi qu'une option par expression. Il est possible avec cette dernière de définir des durées inter-arrivées basées sur d'autres lois (UNIF(a,b) pour uniforme, TRIANG(a,b,c), etc …). L'option « schedule » permet quant à elle de définir manuellement les durées. Nous choisissons ici une loi constante de paramètre 1.

On peut aussi simuler un nombre maximal d'arrivées (entities per arrival) par exemple 20.

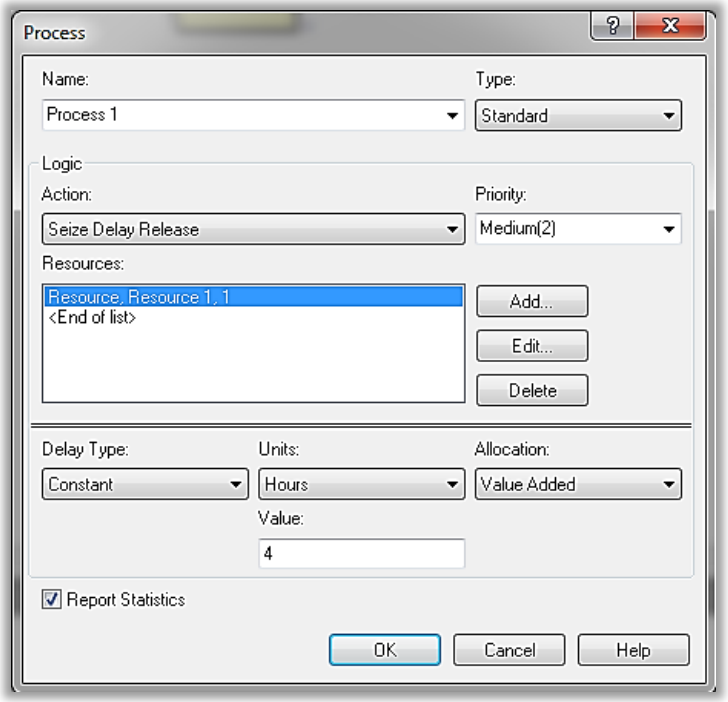


Paramétrage du serveur

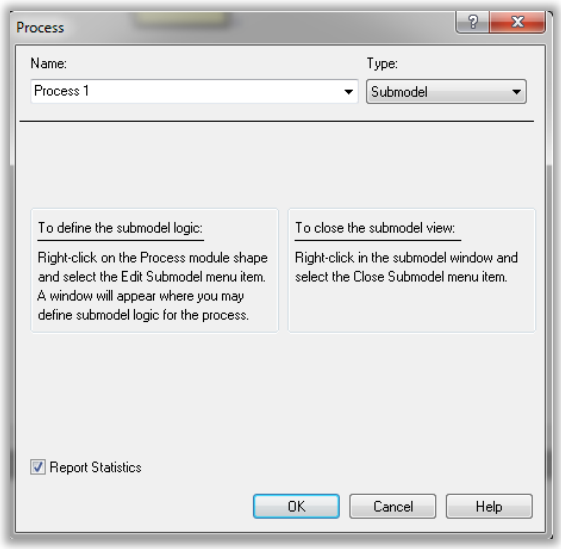
On rentre de la même manière dans les paramètres du bloc « Process ». Plusieurs actions sont disponibles (Delay, Seize Delay, …). Nous choisissons l'option « Seize Delay Release » qui va monter une entité sur la machine/serveur, lui affecter une durée de traitement, et libérer la machine une fois le process terminé. Il faut sélectionner une ressource qui est attachée au process. Si l'on fait « add » une ressource par défaut est générée en quantité unitaire. Nous ne changeons rien à ce fonctionnement.

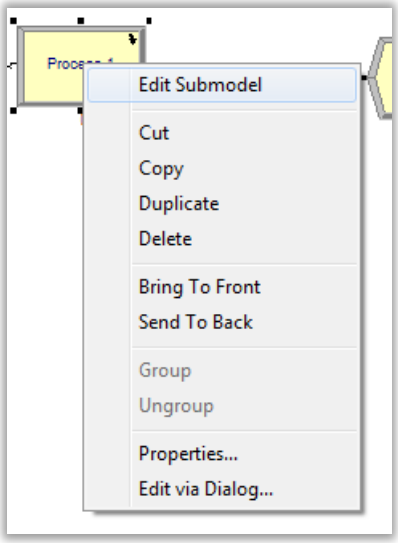


De la zone Delay, il est possible de choisir une loi pour gérer la durée de traitement. Nous sélectionnons une durée constante de paramètre 4.

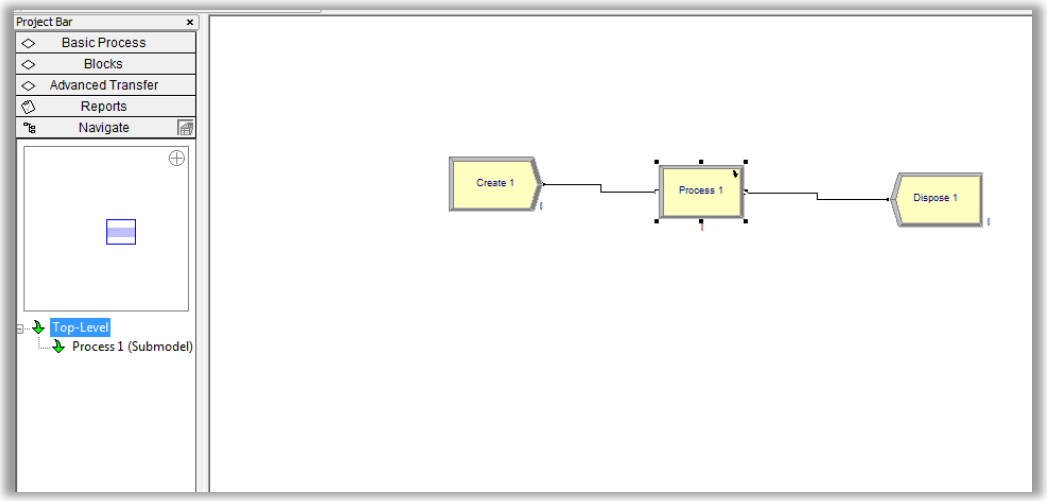


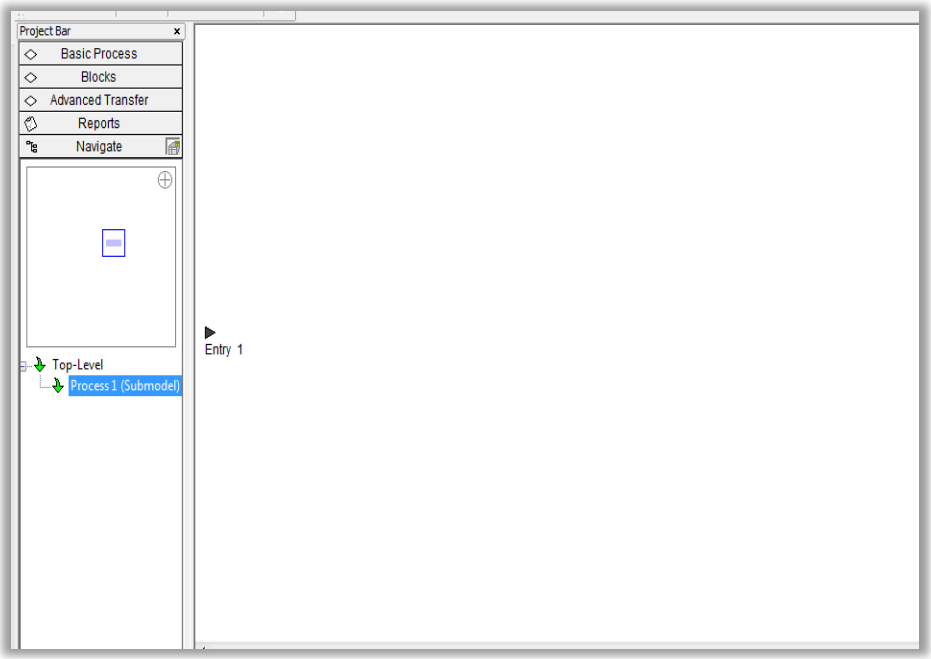
Petit plus : il est possible de créer des logiques de traitement bien particulières en définissant des sous-modèles à partir d'un bloc Process, en choisissant le « Type : Submodel » en haut à droite de la fenêtre.





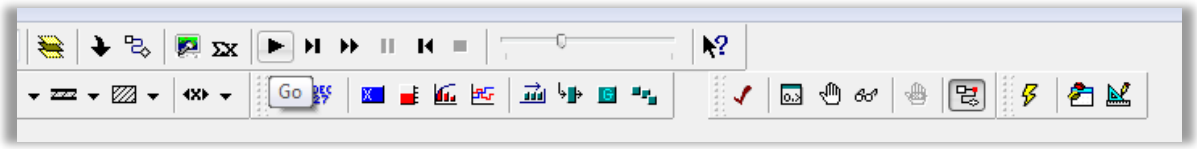
On accède à une nouvelle de zone de modélisation en sélectionnant l'option « Edit Submodel ». Cette zone a une *entrée* et une *sortie* propres et il est alors possible de faire suivre à l'entité un chemin particulier, qui peut lui-même être composé de Processus par exemple. Cette option est une des manières de préserver la lisibilité de modèles complexes en procédant par « niveaux ». On peut alors naviguer dans les différents niveaux via l'onglet Navigate sur la gauche.



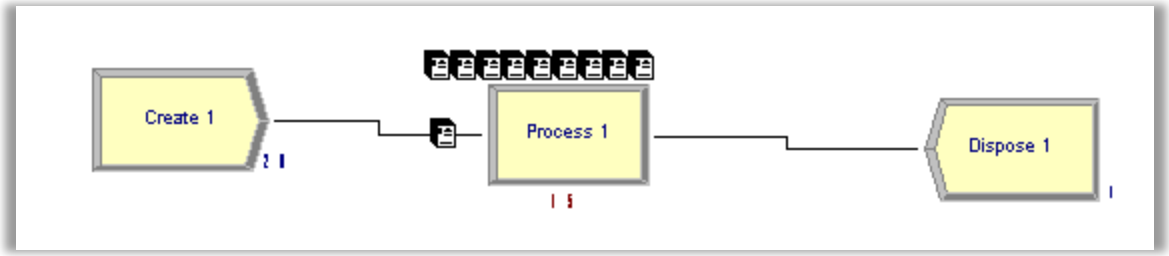


Exécution du modèle

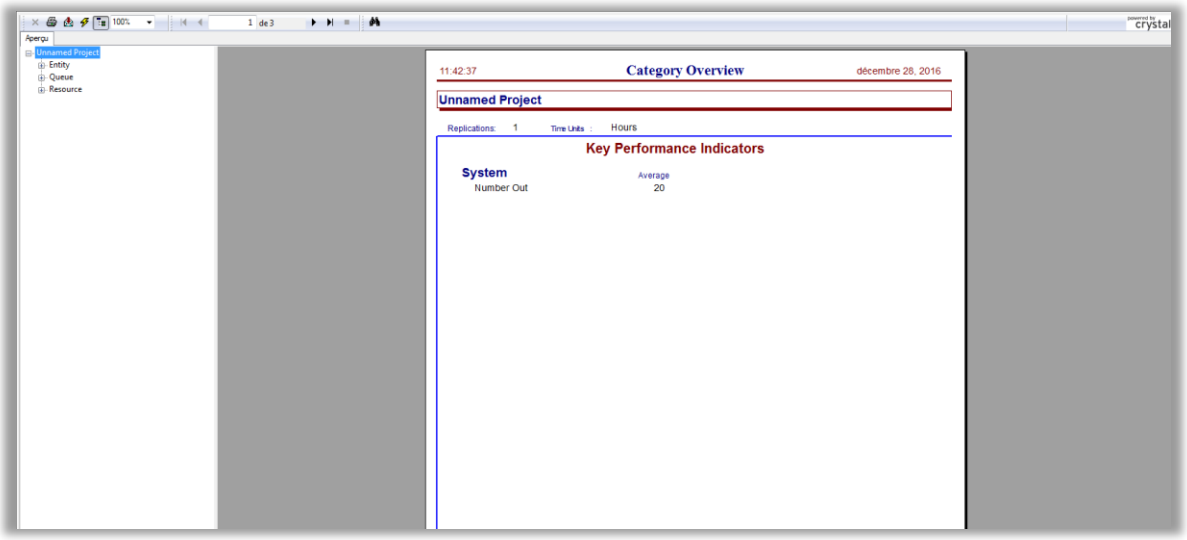
L'exécution se fait via l'interface par l'intermédiaire du bouton Go.



Le modèle offre alors une animation simple (il est possible de changer le visuel des entités, nous verrons cela plus tard). On peut accélérer ou ralentir la vitesse de simulation avec le petit curseur à droite du bouton ‘stop’.



Les résultats apparaissent dans le rapport détaillé. Ce rapport est « dynamique » et il est possible d’accéder à certaines informations particulières via le menu déroulant sur la gauche.



Replications: 1 Time Units : Hours

Entity

Time

VA Time	Average	Half Width	Minimum Value	Maximum Value
Entity 1	4.0000	(Insufficient)	4.0000	4.0000

NVA Time	Average	Half Width	Minimum Value	Maximum Value
Entity 1	0.00	(Insufficient)	0.00	0.00

Wait Time	Average	Half Width	Minimum Value	Maximum Value
Entity 1	28.5000	(Insufficient)	0.00	57.0000

Transfer Time	Average	Half Width	Minimum Value	Maximum Value
Entity 1	0.00	(Insufficient)	0.00	0.00

Other Time	Average	Half Width	Minimum Value	Maximum Value
Entity 1	0.00	(Insufficient)	0.00	0.00

Total Time	Average	Half Width	Minimum Value	Maximum Value
Entity 1	32.5000	(Insufficient)	4.0000	61.0000

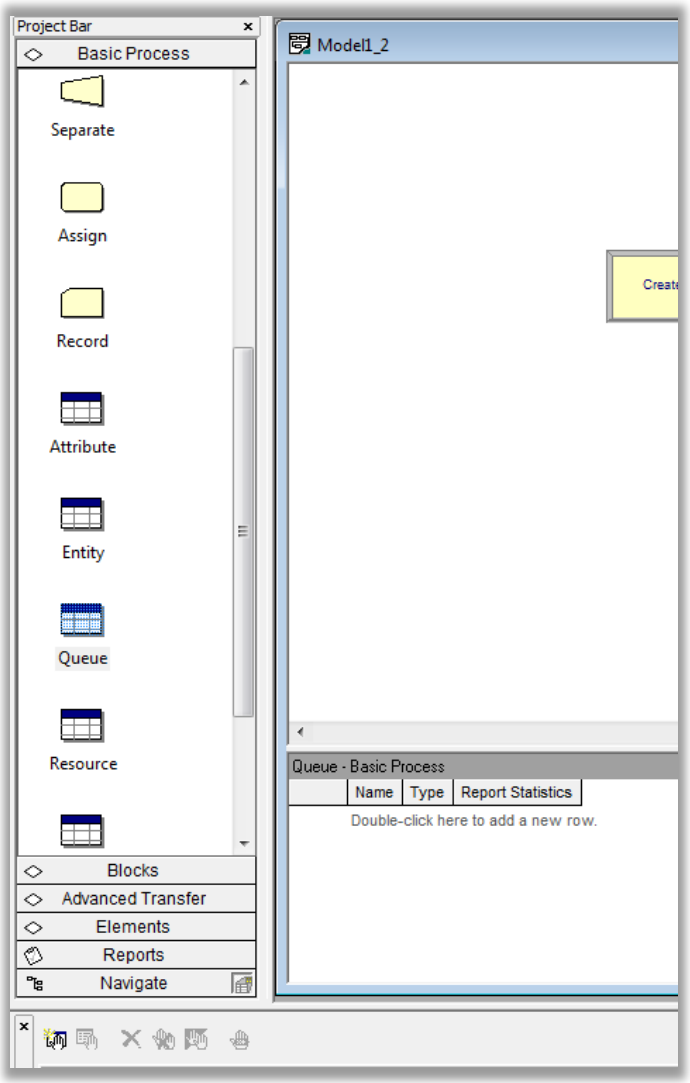
Other

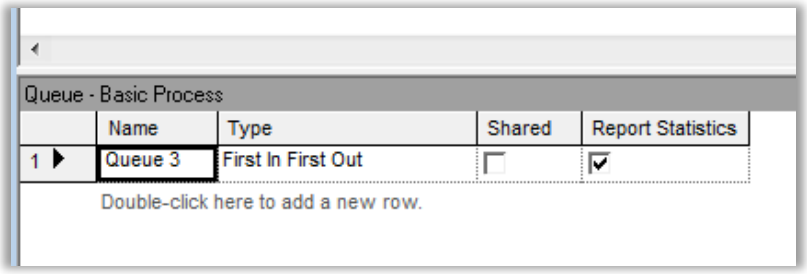
Number In	Value
Entity 1	20.0000

Number Out	Value
Entity 1	20.0000

2.2. Buffers de capacité finie

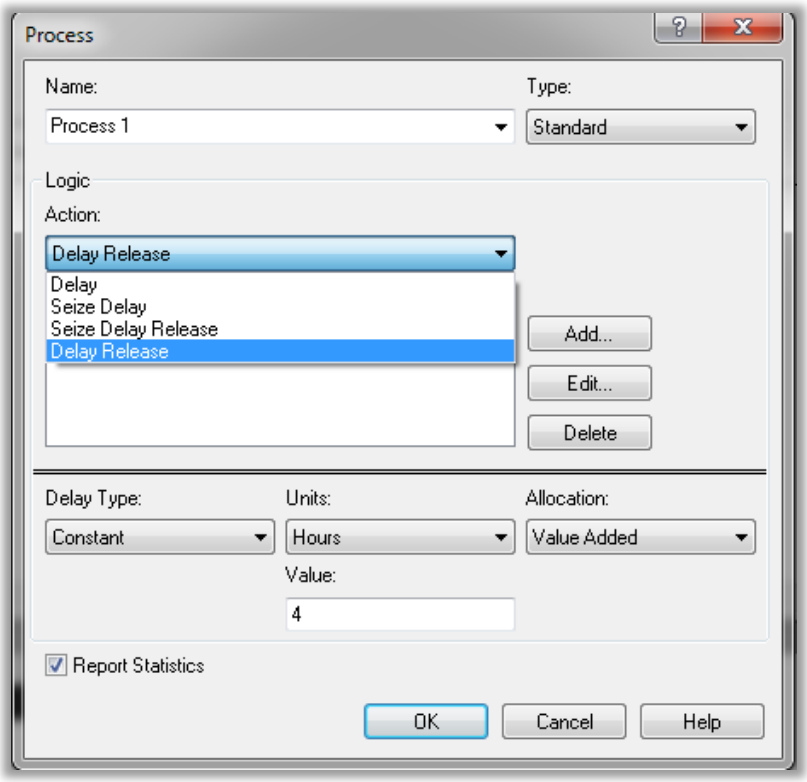
Les fonctionnalités évoquées ci-dessus concernent les Basic processes. Si l'on souhaite définir une file de taille finie, il faut alors modifier le modèle précédemment construit. Dans un premier temps, il faut définir une file à part entière. Pour ce faire, il faut aller dans la section Queue de la zone Basic Process :



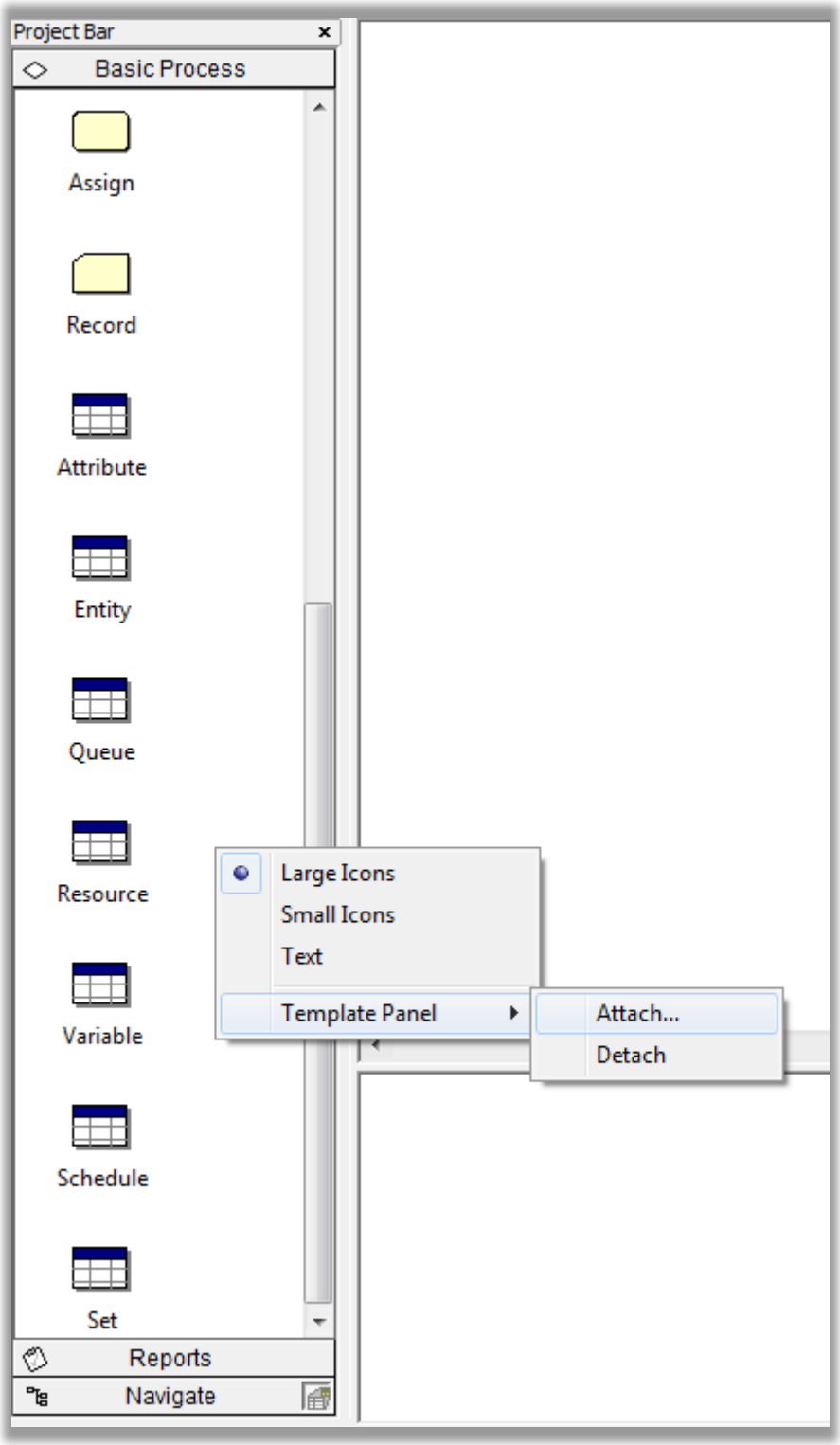


En double cliquant sur la zone concernée, une file est automatiquement générée.

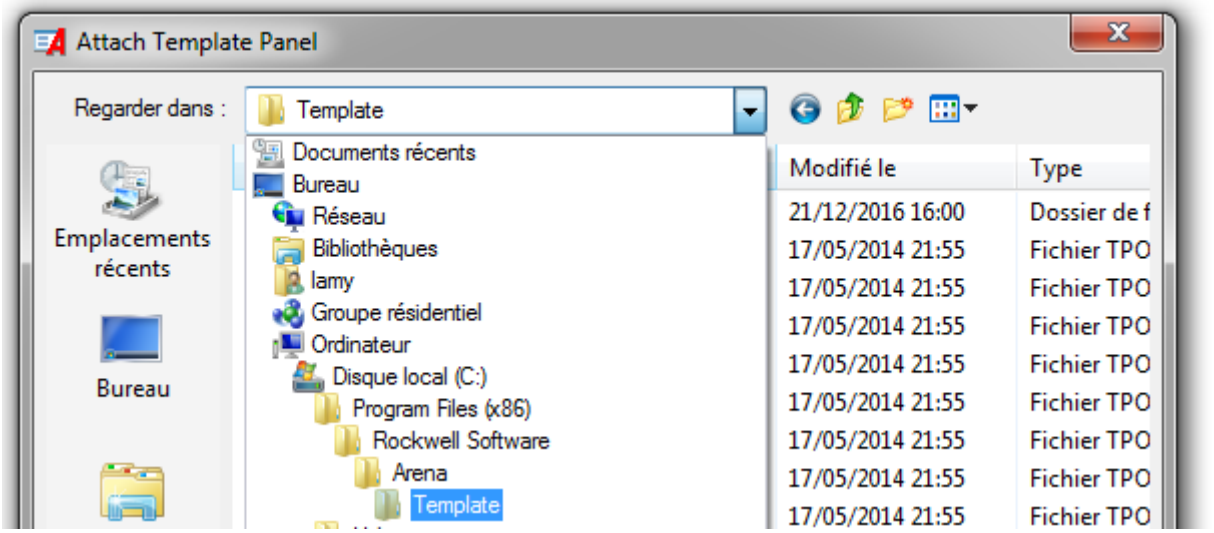
Ensuite, il faut modifier le processus définit dans la première partie en processus « Delay-Release ».



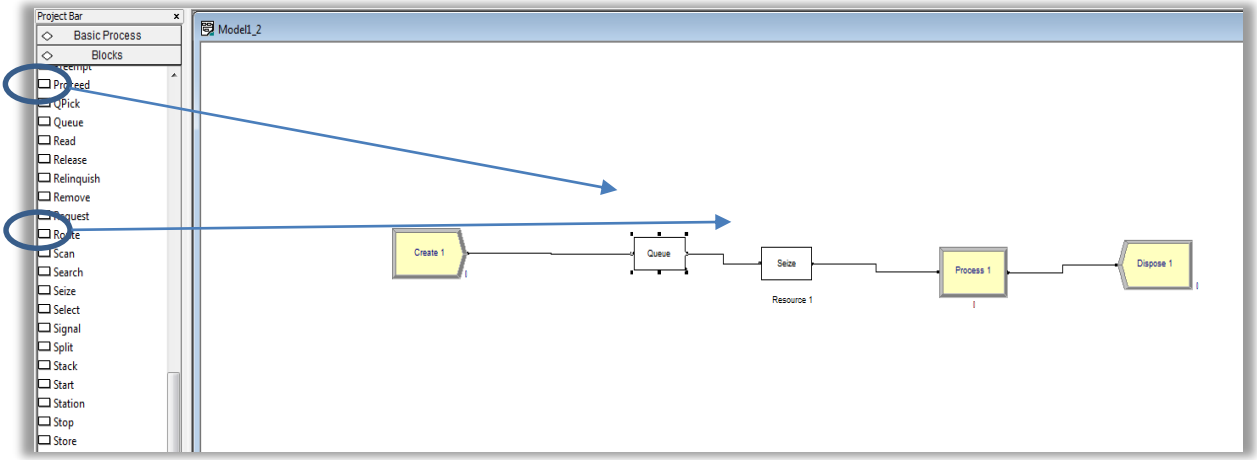
La suite consiste à ajouter une file d'attente spécifique, ainsi que la fonctionnalité seize que nous avons supprimée dans le processus. Pour ce faire, il faut avoir accès à des blocs particuliers du menu déroulant sur la gauche. Nous allons ajouter ces blocs. Il suffit d'aller dans l'option « attach » de la fenêtre ouverte dans la zone « Project Bar ».



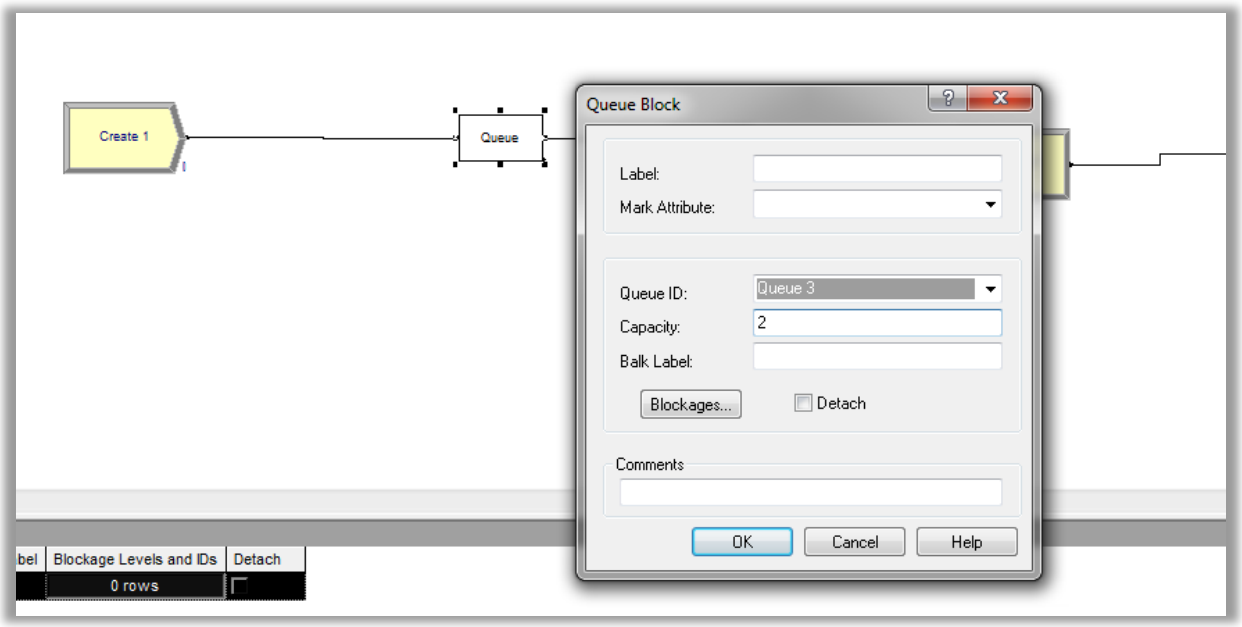
Il faut ensuite ajouter la librairie Blocks.tpo située dans le dossier Template accessible comme dans la figure suivante :

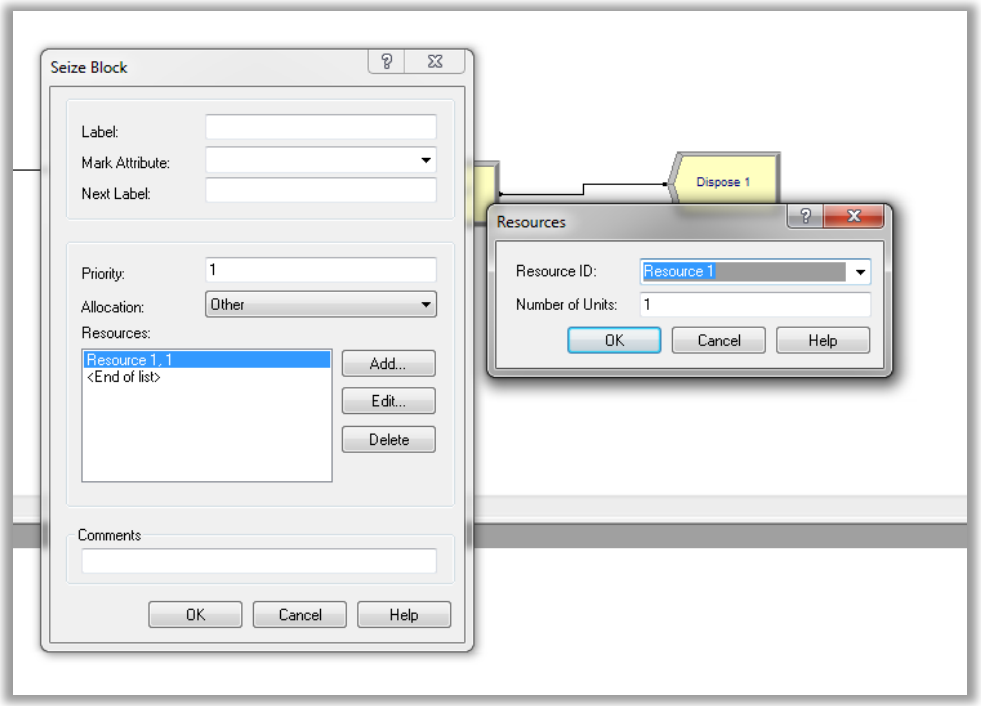


Une fois la librairie attachée, nous pouvons ajouter un bloc Queue et un bloc Seize :

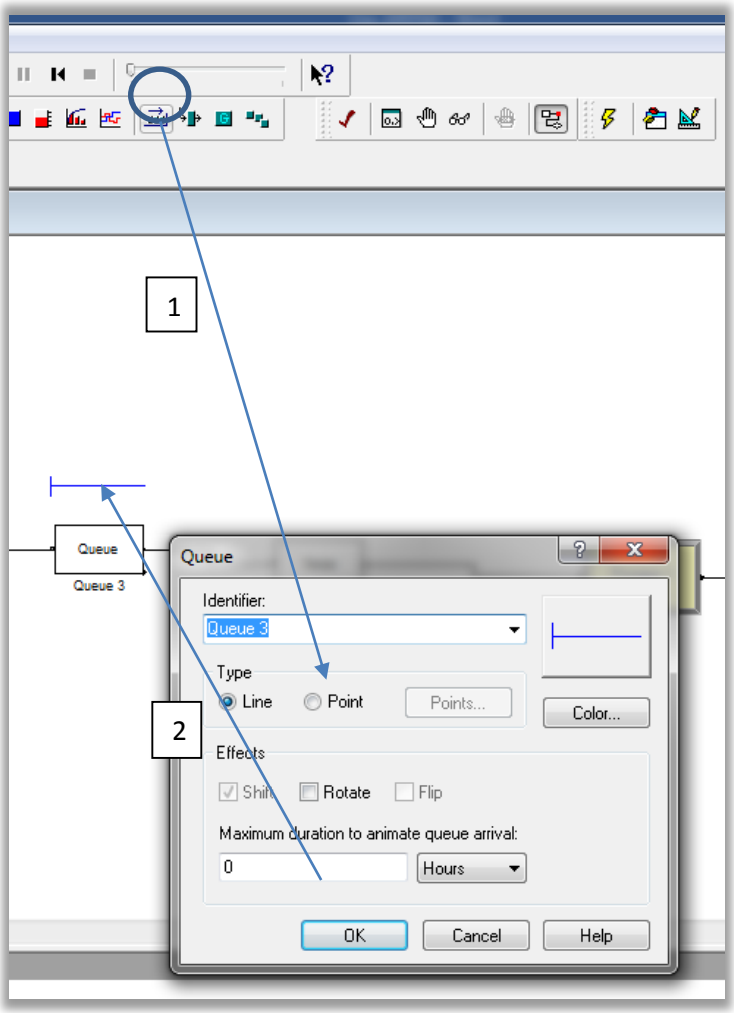


Il faut alors entrer dans la file ajoutée afin de définir que c'est celle que nous avons créée à l'étape précédente. Et préciser que le Seize va être lié à la ressource 1 :

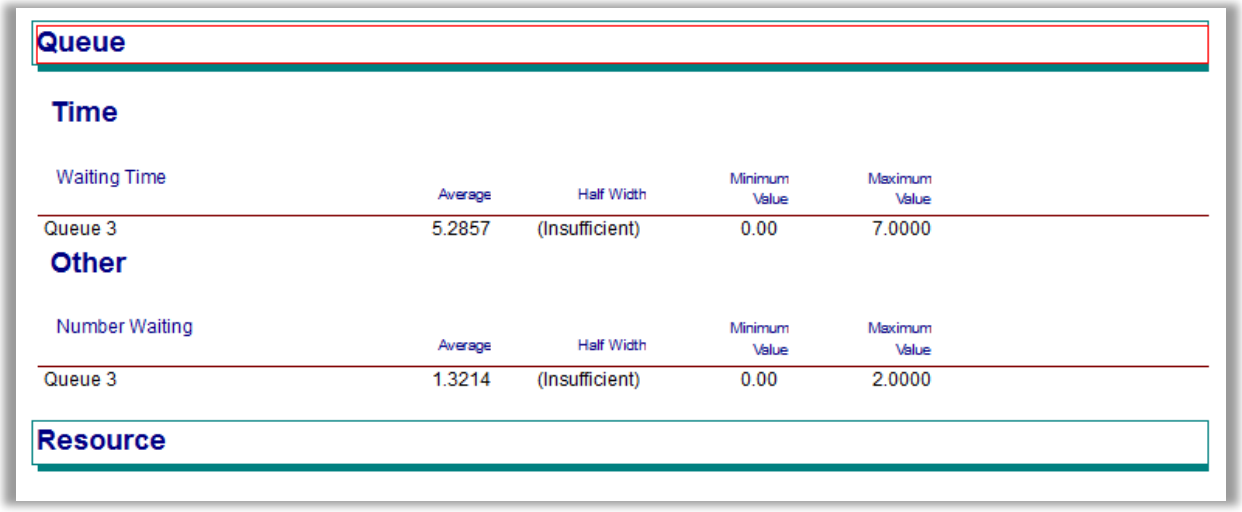




On peut alors relancer l'exécution. Pendant celle-ci, on peut visualiser que le nombre de pièce dans la file est bien limité à 2 pièce. Pour ce faire, il faut ajouter une queue « visuelle » en allant dans l'onglet « queue » en haut. Une fois que la file désirée est sélectionnée, un pointeur s'affiche, il faut alors dessiner la file d'attente (au niveau du bloc « queue » par exemple).



On peut alors visualiser dans les résultats, que le nombre de pièce dans le buffer ne dépasse pas 2 pièces.



Queue

Time

Waiting Time	Average	Half Width	Minimum Value	Maximum Value
Queue 3	5.2857	(Insufficient)	0.00	7.0000

Other

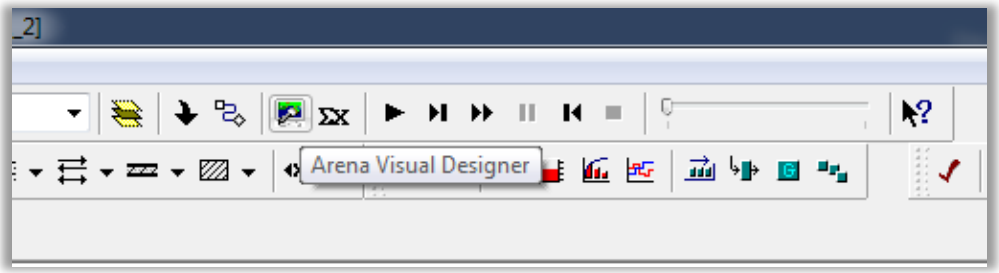
Number Waiting	Average	Half Width	Minimum Value	Maximum Value
Queue 3	1.3214	(Insufficient)	0.00	2.0000

Resource

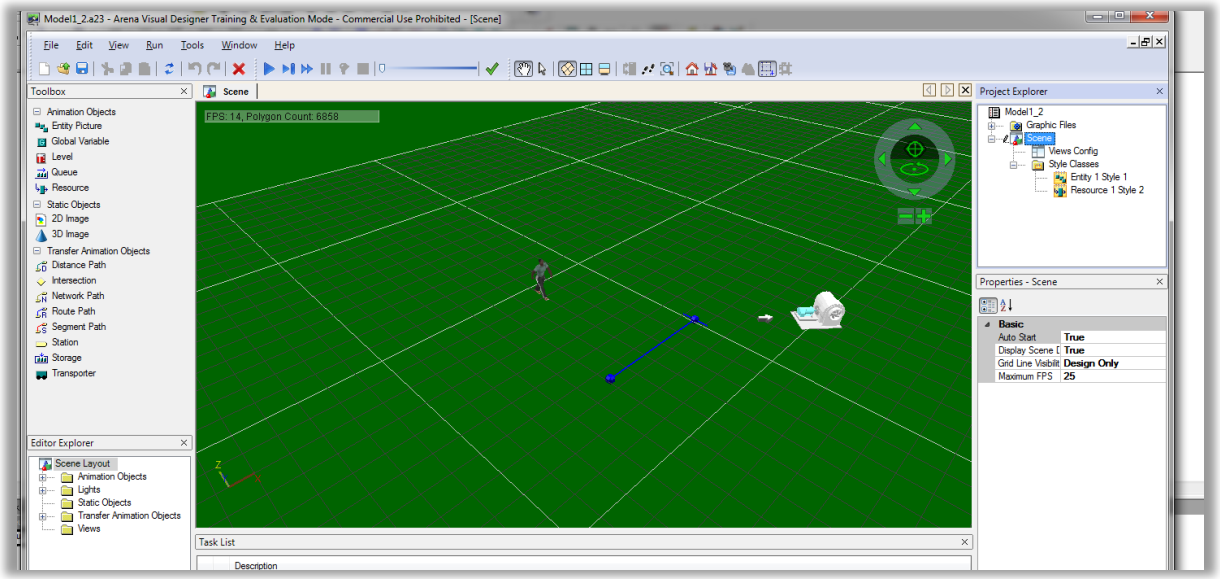
2.3 Simuler une file MM1 avec une visualisation améliorée

Il est possible de modifier le visuel des entités et objets du système. Pour ce faire, il faut aller sur le petit tableau « Entity » de la section « Basic Process ». Une entité est déjà définie (Entity 1). On peut changer l'image associée à cette entité. Pour avoir une femme en peignoir rouge, il suffit de sélectionner Picture.Woman dans le menu déroulant correspondant, oui oui ...

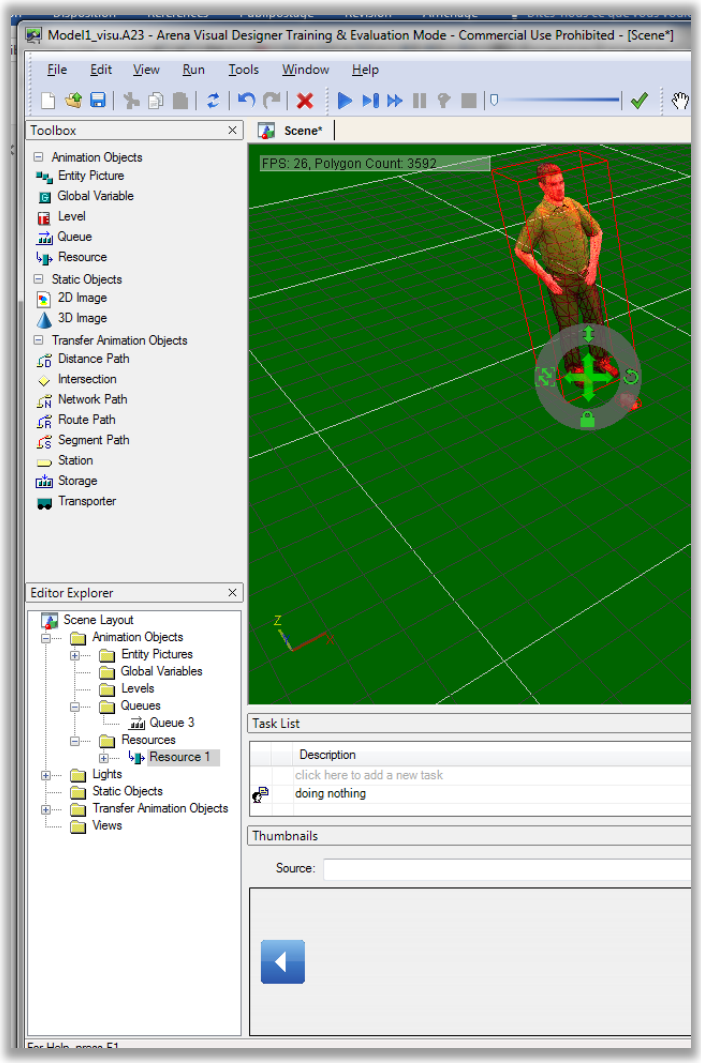
Il est possible d'avoir une vue en 3D également via « Arena Visual Designer ». Avant cela, il faut s'assurer qu'une image est bien associée à l'entité dans Arena. Ensuite il faut aller sur l'icône correspondante :



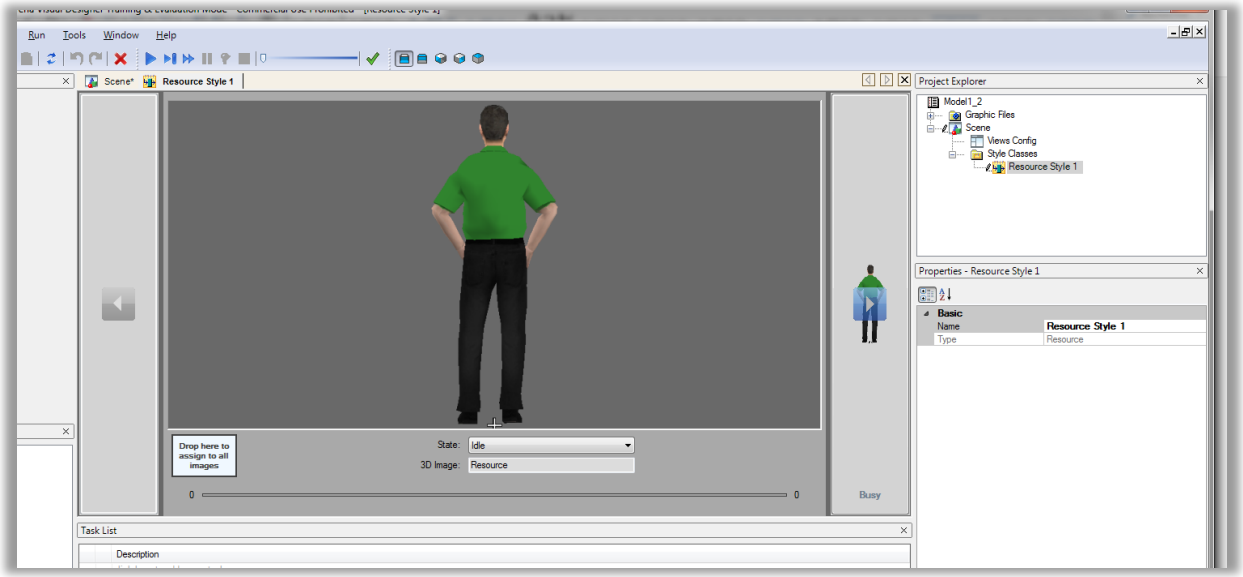
On arrive alors sur une zone de dessin :



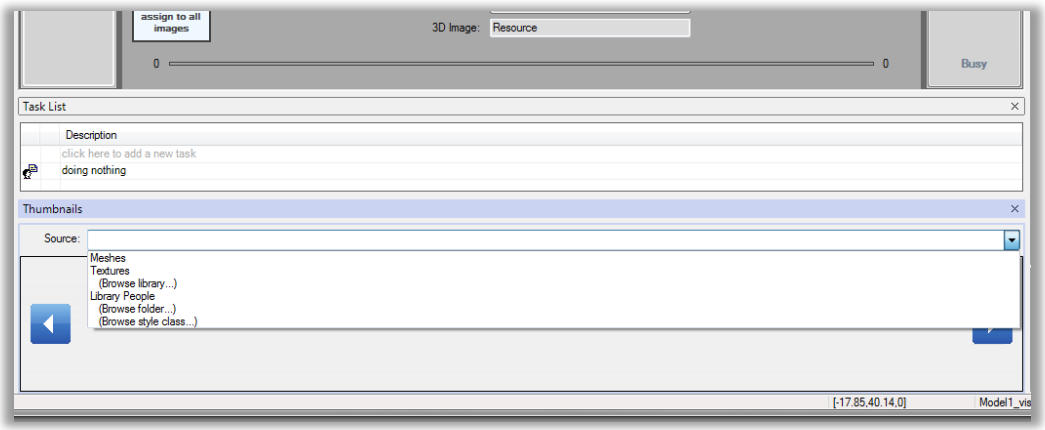
Commençons par déposer un serveur. Un humain pour cet exemple. Il suffit de faire glisser sur la zone de dessin une ressource (depuis la toolbox à gauche).

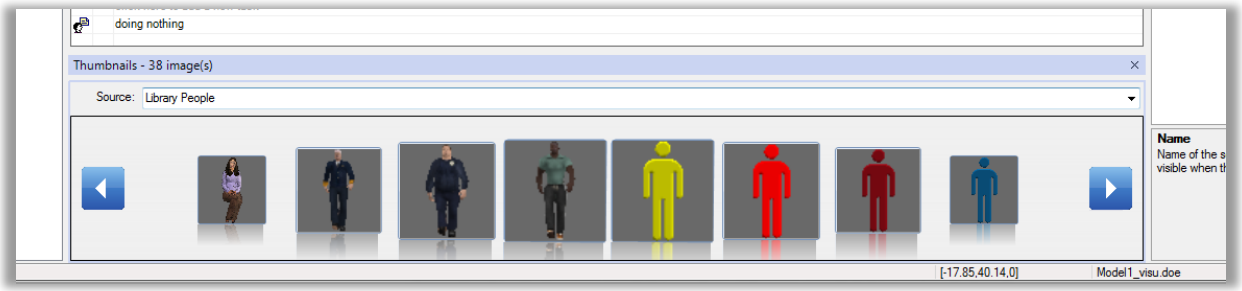


Ensuite, en double-cliquant sur la ressource ajoutée, on arrive dans l'éditeur de ressource :

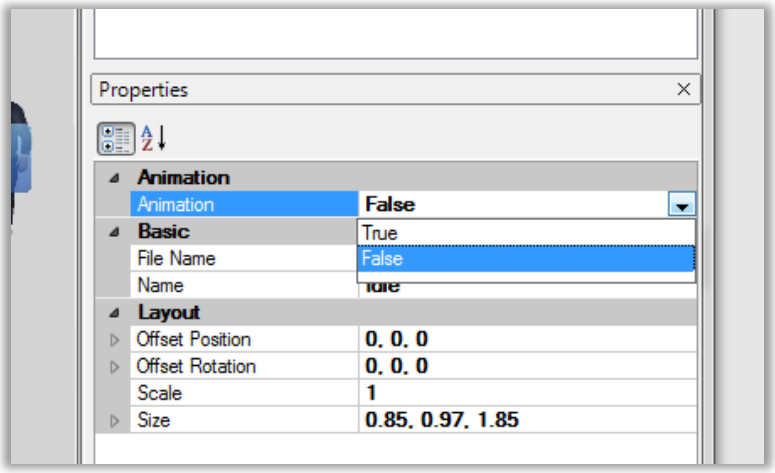


L'image par défaut est une image statique, nous allons la modifier pour choisir un personnage dynamique. Pour ce faire, il faut aller dans la zone « Thumbnails » en bas, et ajouter la source library People :

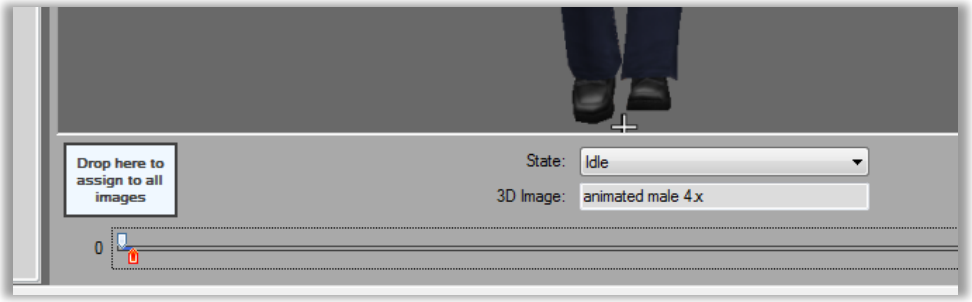




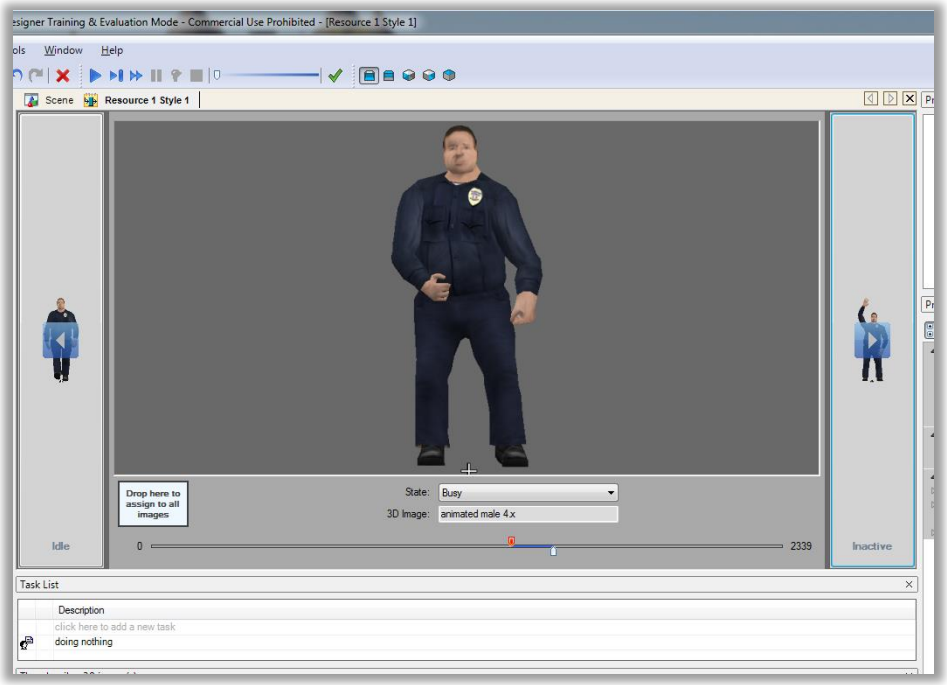
Les trois bonhommes avant le personnage jaune sont des bonhommes dynamiques. Nous sélectionnons l'officier de police. Il suffit de faire glisser l'officier vers le carré « drop here to assign to all images ». Une fois cela fait nous pouvons définir la gestuelle du personnage en activant dans l'onglet « animation » (sur la droite) l'option 'True'.



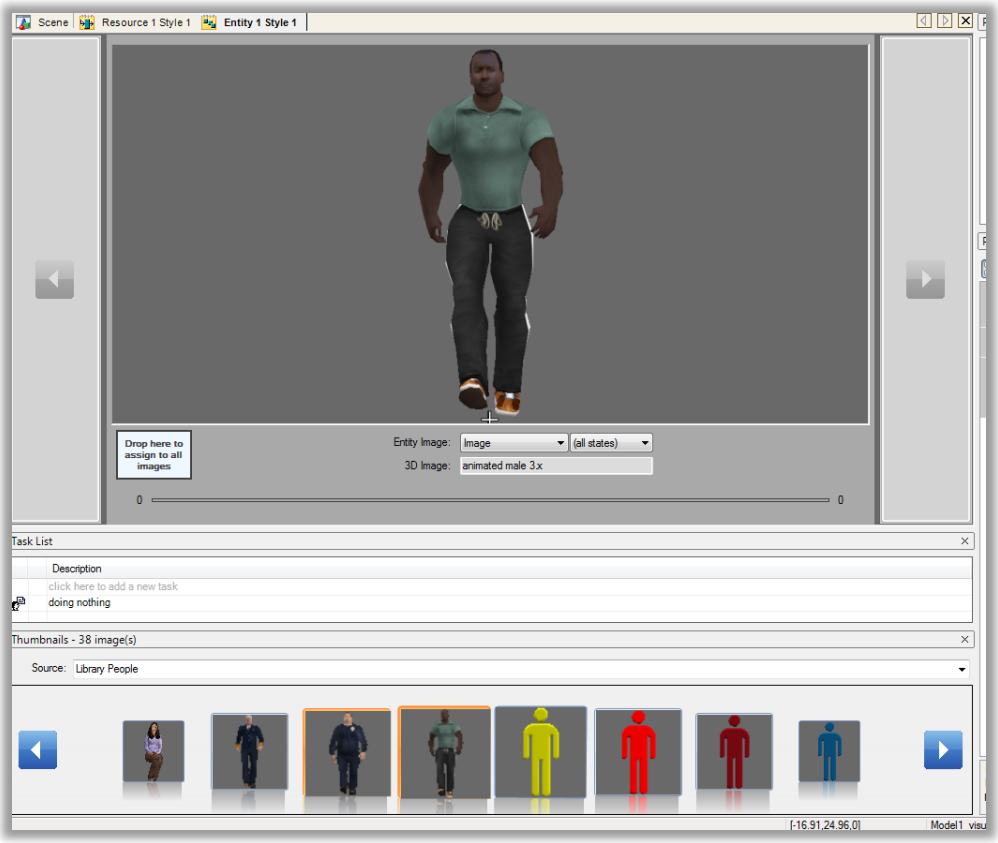
Une barre défilante est alors utilisable sous le personnage :



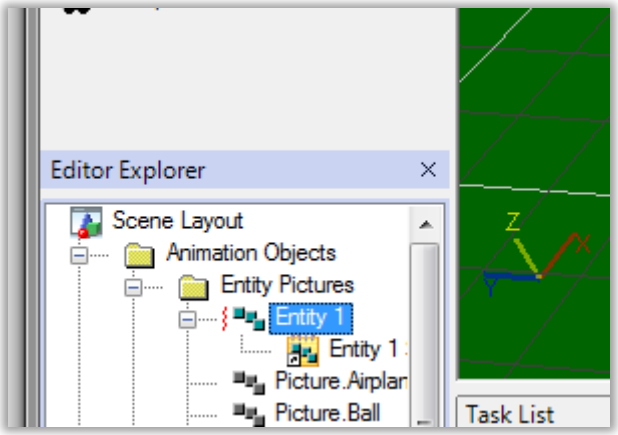
A partir de cette barre défilante nous pouvons choisir certaines actions effectuées par l'officier. On peut choisir les premières secondes pour l'état « Idle ». pour l'état « Busy » on peut prendre une autre zone, ainsi que pour l'état « Inactive » :



Nous retournons sur la zone de dessin « Scene ». Nous allons maintenant ajouter une entité. De la même manière nous faisons glisser un objet « Entity Picture » sur la zone de dessin. Un magnifique carton est alors crée. Double cliquons sur le carton. Nous faisons comme pour l'étape précédente, à savoir faire glisser sur la zone « drop here » le dessin d'un personnage, celui juste à gauche du bonhomme jaune par exemple.

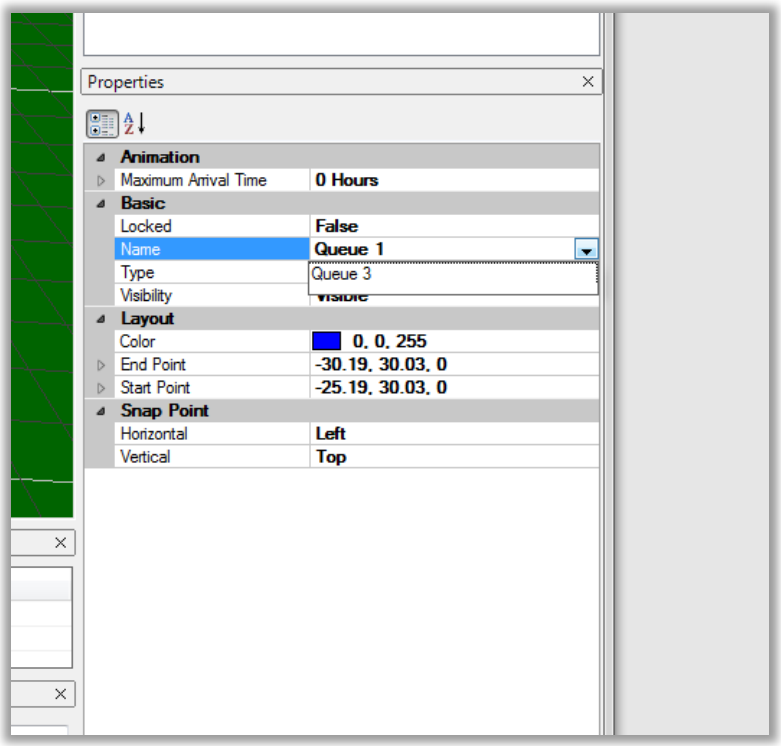


Nous retournons sur la zone de dessin. Nous voyons une petite vague rouge à côté de l'entité dans la zone « editor explorer ».

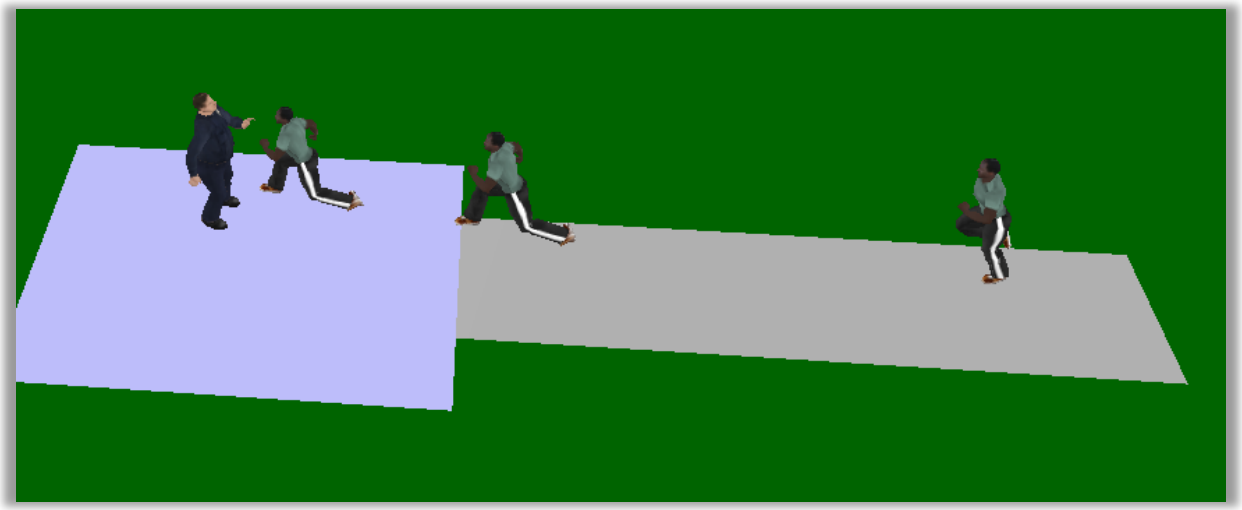


Cela veut dire que l'entité n'est pas utilisable en l'état. Pour faire le lien entre Arena et le visual designer. Pour ce faire, il faut sélectionner le bonhomme correspondant à l'entité et modifier le nom « Entity 1 » en « Picture.woman » dans le menu déroulant. Oui, ce n'est pas logique. Nous aurions évidemment pu sélectionner un personnage feminin, mais l'important est de montrer que le lien est effectué sur l'image associée à l'entité dans Arena indépendamment de son 'type'.

Maintenant il nous reste à ajouter la file d'attente. Il suffit de glisser/déposer comme précédemment une « queue » sur la zone de dessin. Dans le cas où plusieurs queues seraient définies, il suffit de sélectionner la bonne dans le menu déroulant :

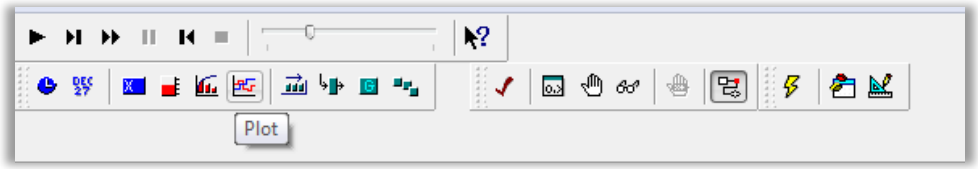


Enfin il est possible de définir le temps de transition dans la file d'attente via l'option Maximum Arrival Time. Si cette option est à 0, les entités seront déposées directement les unes derrière les autres. En fixant cette durée à 15 secondes, on peut voir les entités bouger avant d'arriver à leur position définitive dans la file d'attente. On peut également disposer grossièrement des zones de dessin pour délimiter la zone dans laquelle la simulation prend place via le bouton « 2D image » de la ToolBox. Lorsqu'on lance la simulation nous obtenons une image comme suit :

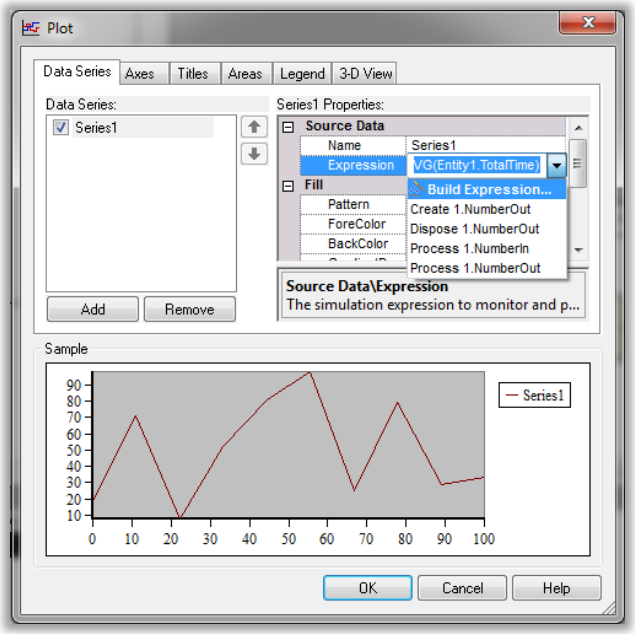


2.3) Visualisation dynamique d'information

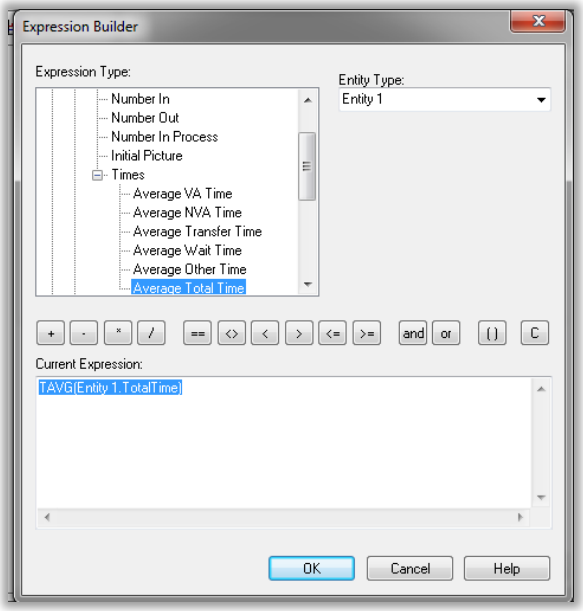
Pour ajouter une courbe dynamique en fonction de l'avancement de la simulation, il faut aller sur l'onglet « Plot ».



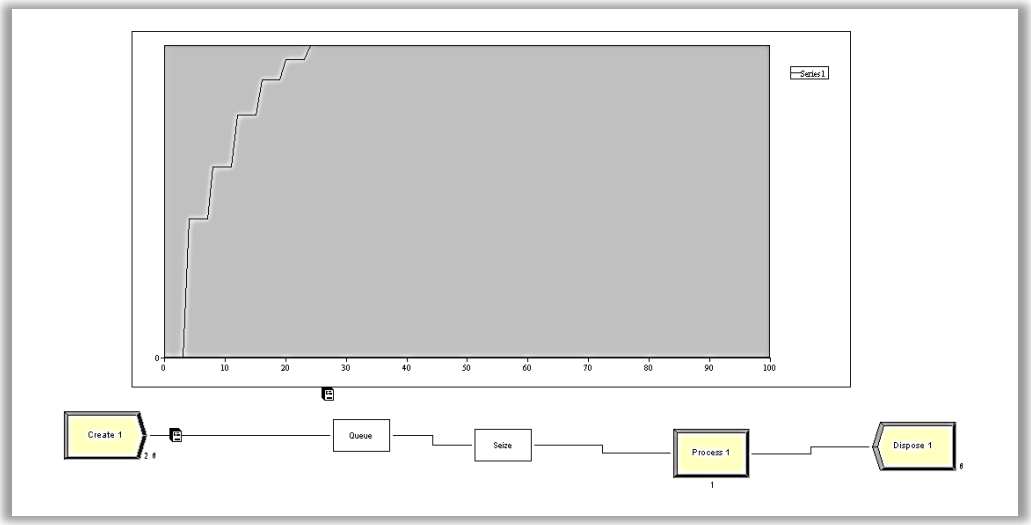
Une nouvelle fenêtre s'ouvre. Aller sur le bouton 'Add'. Une nouvelle série est ajoutée. Il est possible de définir l'expression qui est visualisée. Dans la liste déroulante associée, sélectionner « Build Expression ».

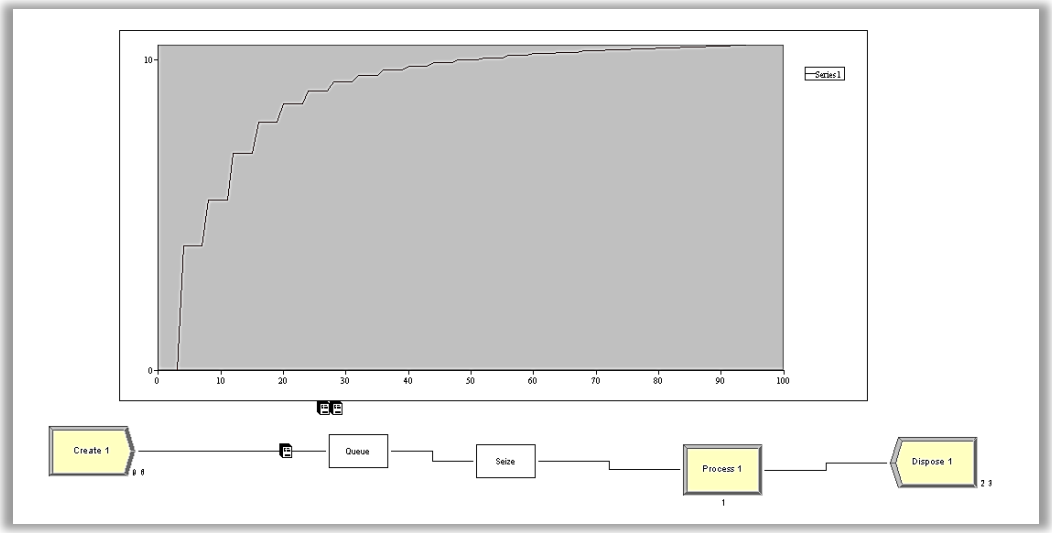


Ici, nous pouvons sélectionner plusieurs expressions 'types', ou en définir de nouvelles. Pour l'exercice, nous prenons le temps moyen total passé dans le système par une entité :



Après avoir validé tous les choix, il suffit de mettre la zone de graphique à côté du modèle. Lorsqu'on lance la simulation, le graphique évolue au fil du temps :

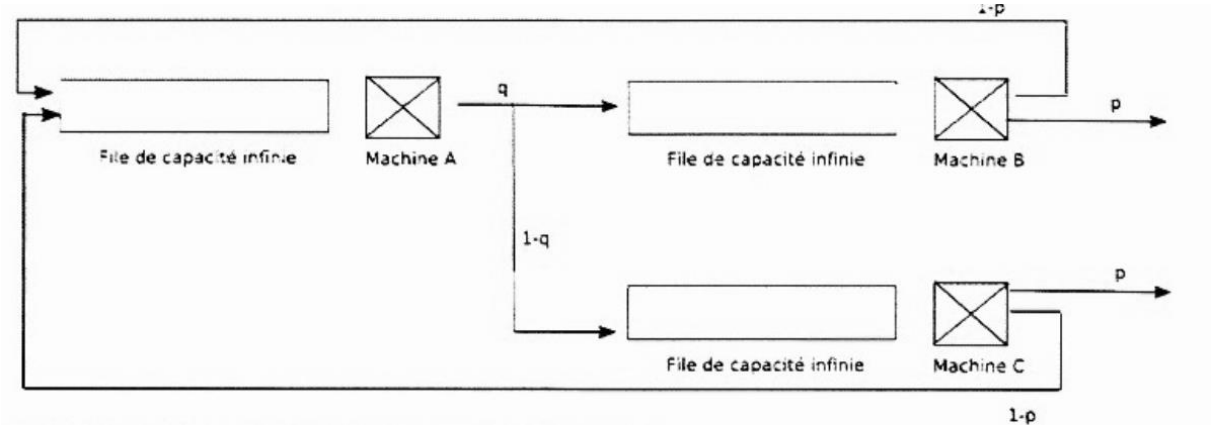




3) Simuler un système avec routage probabiliste

La durée inter-arrivée ainsi que les temps de traitement suivent une loi constante de paramètre :

- **Lam** pour la durée inter-arrivée
- **Sa** pour la durée de traitement sur la machine A
- **Sb** pour la durée de traitement sur la machine B
- **Sc** pour la durée de traitement sur la machine C
- Les paramètres **p** et **q** sont des paramètres de transition.

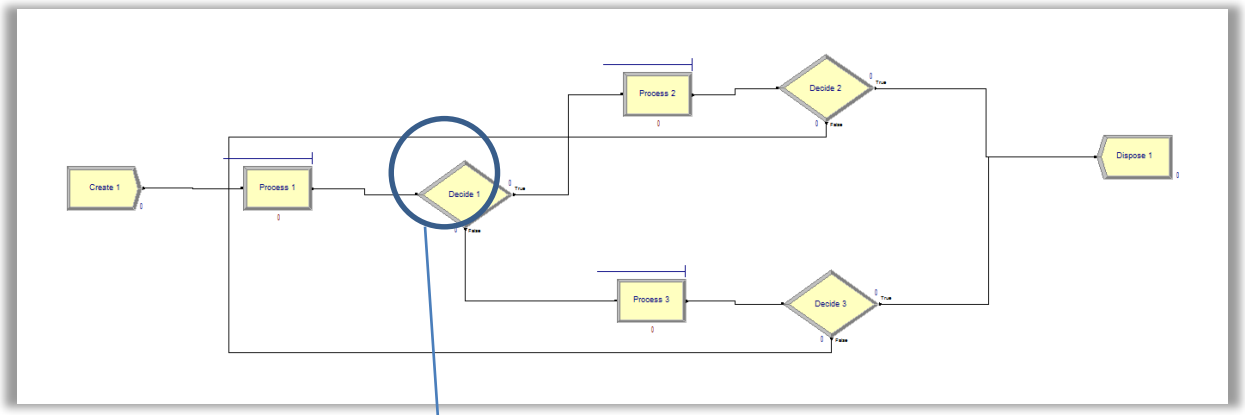


Il s'agit de réaliser un modèle de simulation en prenant par exemple :

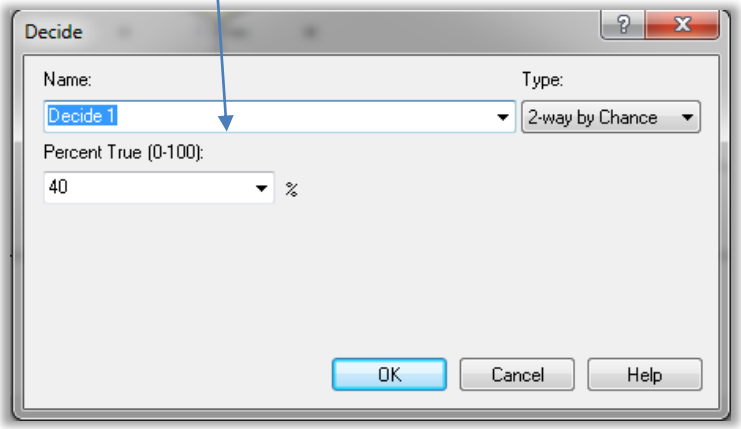
Sa=1, Sb=0.8, Sc=0.6, p=0.6, q=0.4 et Lam=10.

3.1) Réalisation du modèle

En utilisant 1 create, 3 processes et 1 dispose, on peut construire le réseau.



Comme montré dans la figure ci-dessus, un nouveau bloc est présent : le bloc « Decide ». pour chaque decide qui suit un bloc process, il faut définir la probabilité d'aller dans une direction donnée.

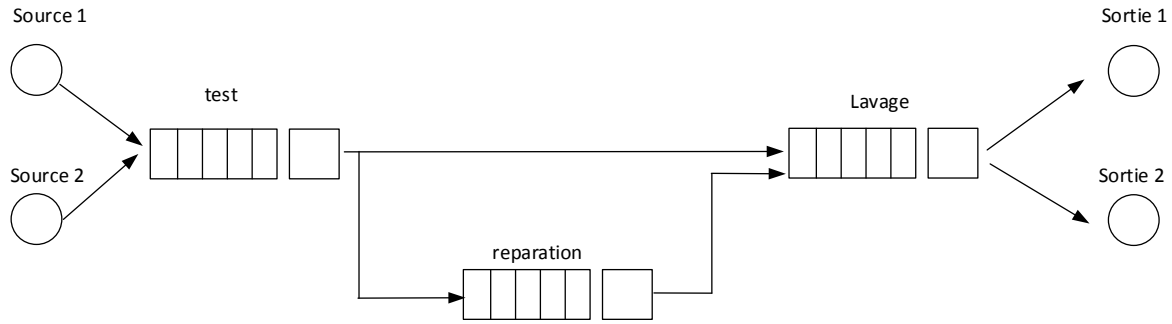


Ici nous sélectionnons pour le 1er Decide le type « 2-way by Chance » avec une probabilité True de 40% (=q). Nous procédons de même pour les autres blocs Decide. Et nous joignons les autres sorties (équivalentes au *Else*) vers les entrées correspondantes (i.e. : la sortie *Non* du bloc 'Decide 2' est associée à l'entrée du processus 1).

4) Simuler un système avec routage probabiliste

La description de ce problème est disponible à l'adresse suivante :

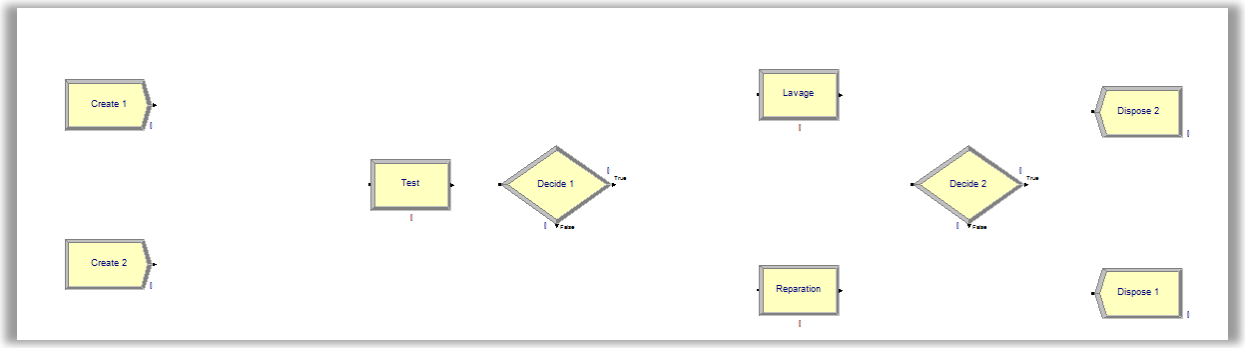
https://www.youtube.com/watch?v=r-VeWcWPhps&list=PLUVWSuhLpltxXkioXlJbCFD6nM9jFma3J



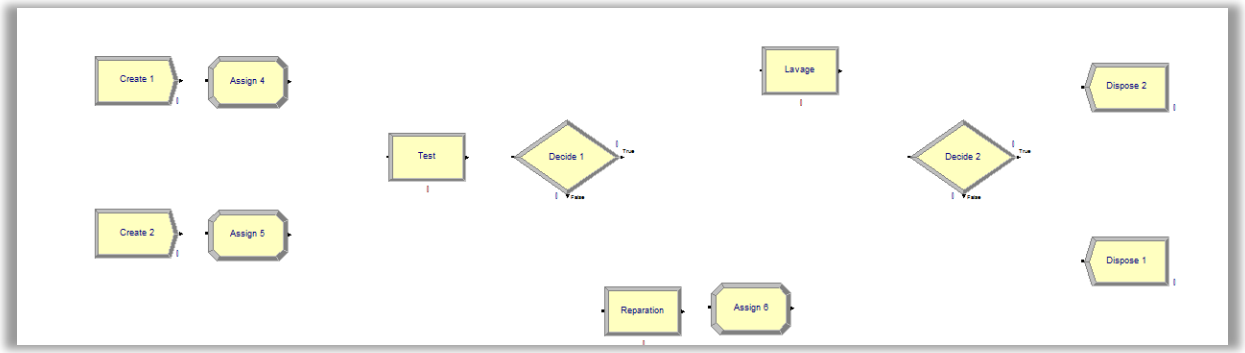
- La durée inter-arrivée sur la source 1 suit une loi exponentielle de paramètre 15 ;
- La durée inter-arrivée sur la source 1 suit une loi exponentielle de paramètre 25 ;
- La probabilité qu'à l'issu du test, on entre dans une phase de réparation est de 20% ;
- Les pièces (véhicules) n'ayant subis aucune réparation sortiront par la sortie 1 et les véhicules ayant subies une réparation par la sortie 2 ;
- Le temps de séjour sur la machine « test » dépend du type de pièce, une loi ……. Pour les pièces de type 1 (Camions) et une loi … pour les pièces de type 2 (Voiture) ;

4.1. Construction du modèle : entités et sources

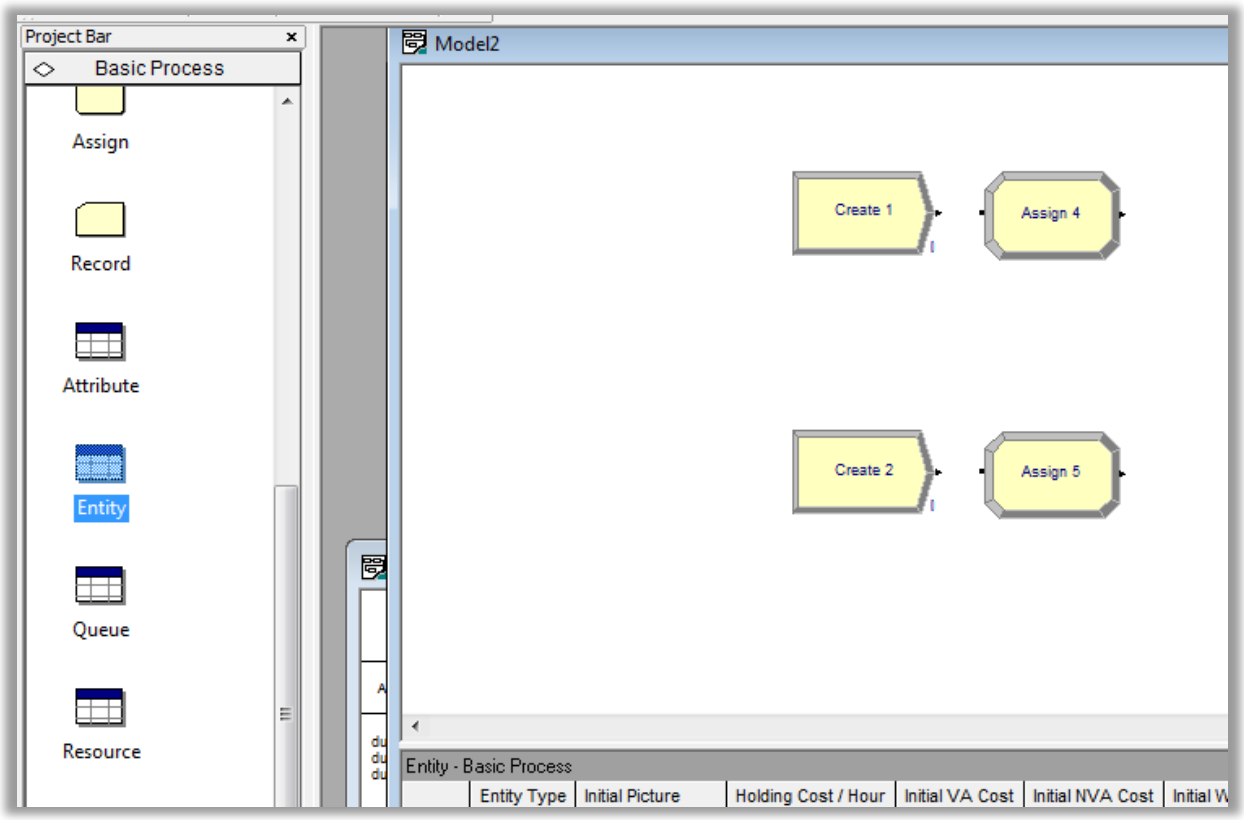
On peut immédiatement placer les différents éléments du modèle.



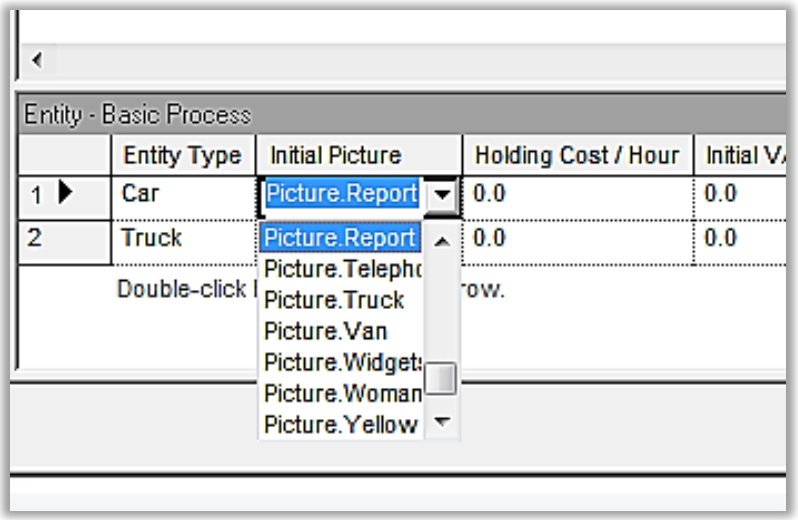
Nous allons ajouter maintenant 3 blocs importants : les blocs « assign ». ils vont permettre de fixer des valeurs aux attributs des entités. Ces blocs sont dans la section basic process.



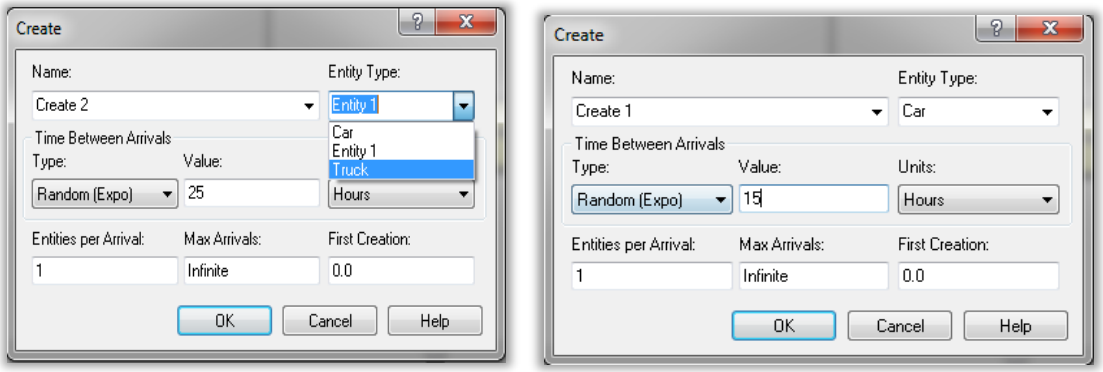
Tous les blocs utiles sont désormais présents. Nous allons maintenant définir deux types d'entités différentes : les camions et les voitures. Pour ce faire il suffit d'aller dans la section entity de la zone « basic process ».



Ensuite il suffit de double cliquer dans la zone indiquée. Nous ajoutons deux types d'entités que nous nommons Car et Truck. Il est possible d'associer des images prédéfinies à ces entités. Nous sélectionnons deux images différentes dans le menu déroulant « Initial Picture » : « Picture.Van » et « Picture.Truck ».

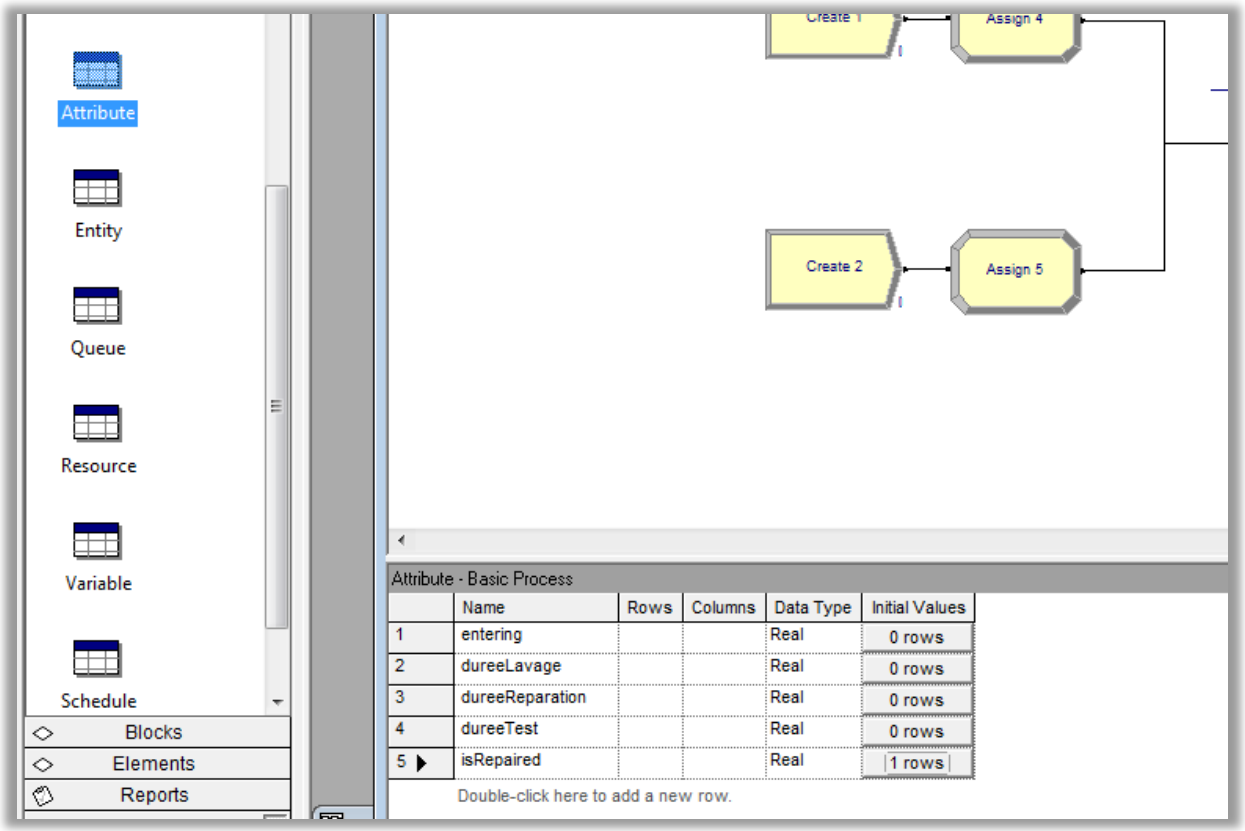


Nous allons ensuite définir quel bloc create gère quel type d'entité :



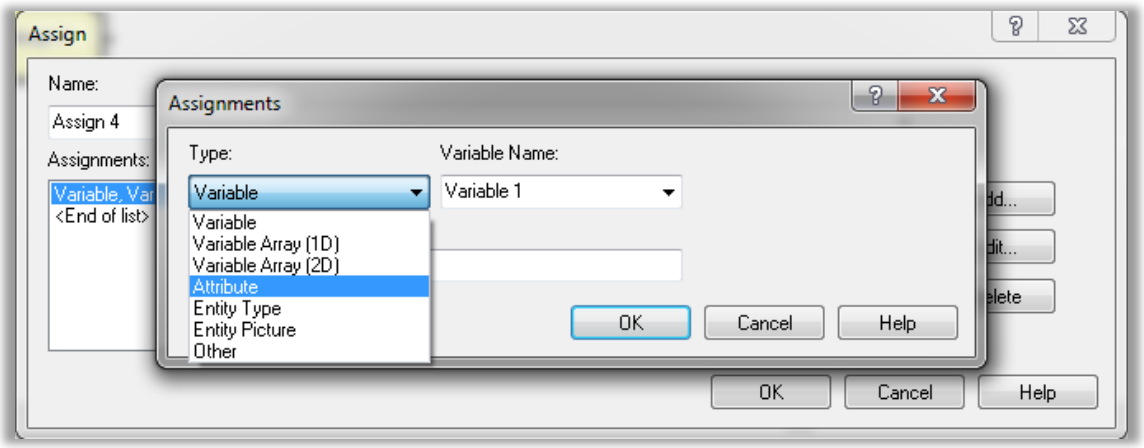
Dans les paramètres du « Create 2 » nous choisissons le type d'entité « Truck ». nous faisons de même avec le « Create 1 ». nous définissons également une loi Exponentielle de paramètre 25 pour les camions, et 15 pour les voitures.

Ces entités vont avoir des temps de traitement différents dans les processus de Test, Lavage et Réparation. Nous allons définir des attributs custom pour les entités. Pour ce faire nous allons dans la zone « Attribute » de la section « Basic process ». Nous ajoutons les attributs suivants :

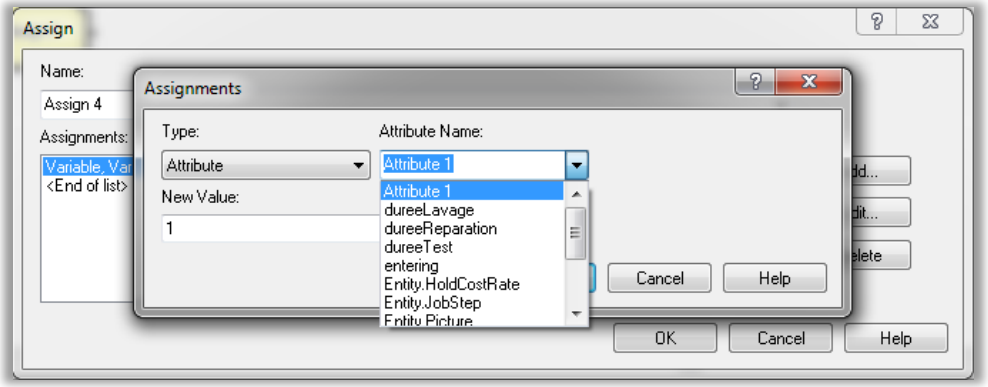


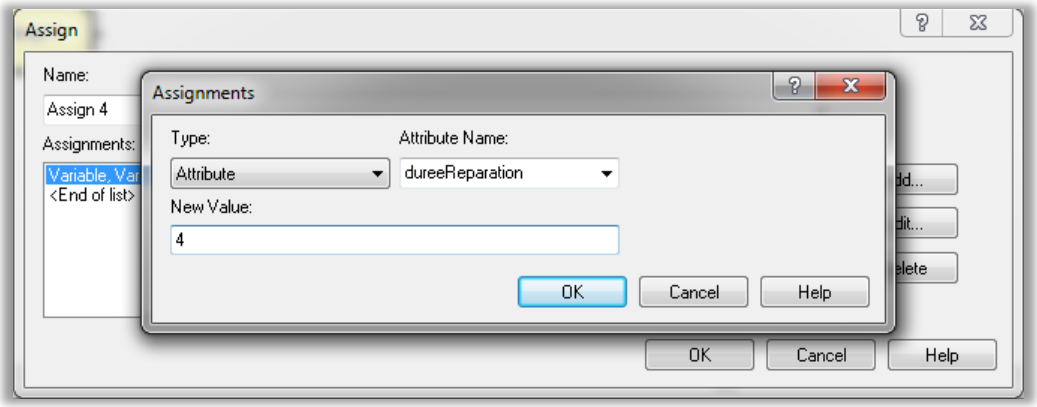
Il suffit de double clicker dans la zone mentionnée et de nommer les attributs. Par défauts ce sont des réels. Il est possible de faire des matrices également en définissant le nombre de lignes/colonnes. Nous avons également ajouté un attribut « isRepaired » pour savoir vers quelle sortie orienter les véhicules ayant subi une réparation. Nous définissons une valeur de 0 pour cet attribut en double cliquant sur l'onglet « Initial Values » correspondant.

Nous allons maintenant définir quelles valeurs prennent ces attributs, en fonction du type de véhicule. C'est là qu'interviennent les blocs « Assign ». Nous allons dans le bloc assign qui suit le Create 1 (voitures). Et nous cliquons sur « add ».



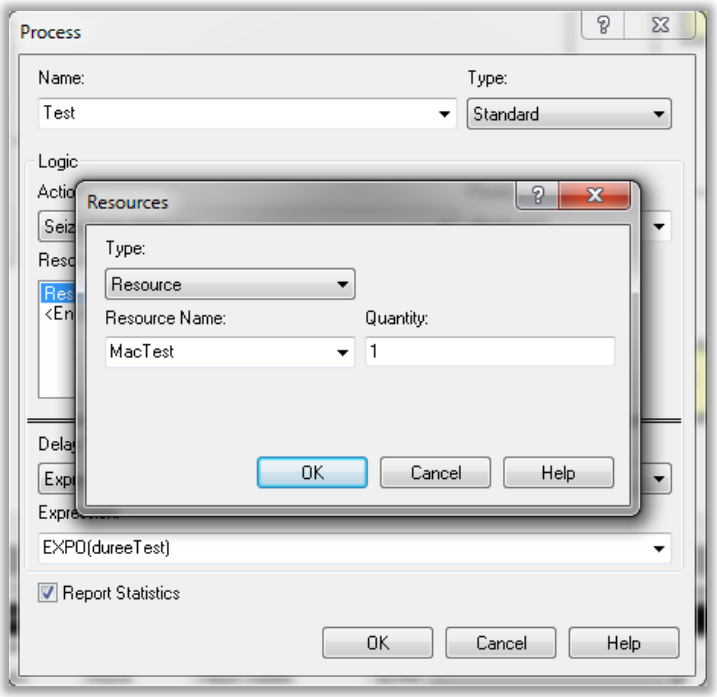
Nous allons ajouter une affectation concernant un attribut. Nous selectionnons l'attribut souhaité, et nous mettons la valeur associée :



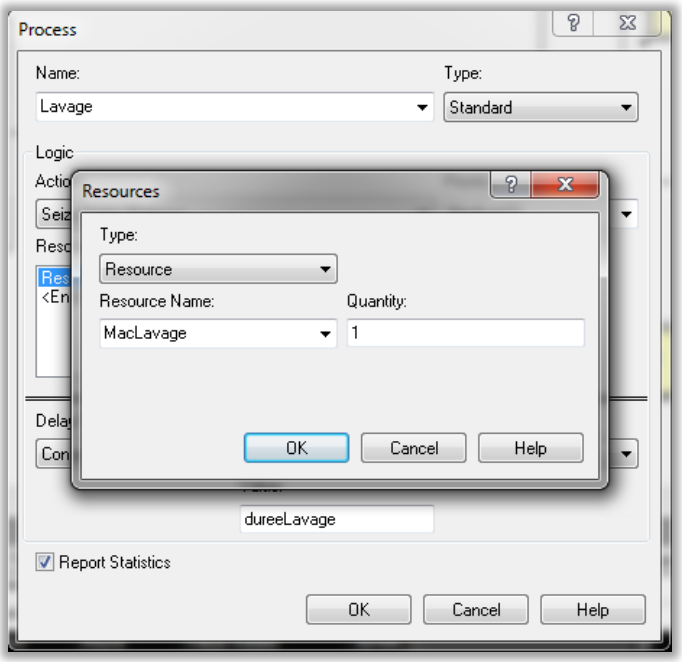


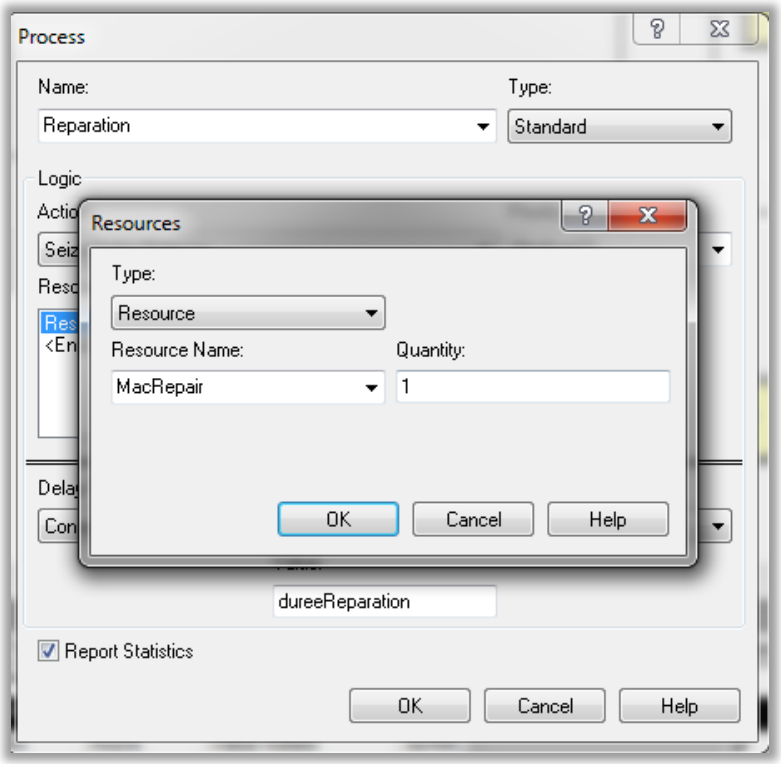
Nous répétons le processus pour les attributs « dureeLavage » et « dureeTest » dans ce bloc assign, mais également dans celui qui suit le deuxième Create (camions).

Nous allons maintenant utiliser les attributs que nous venons de définir pour gérer les temps de traitement au niveau des processus. Nous commençons par le process « Test ». Nous sélectionnons une logique de type « Seize-Delay-Release ». nous ajoutons ensuite une ressource que nous nommons « MacTest ».

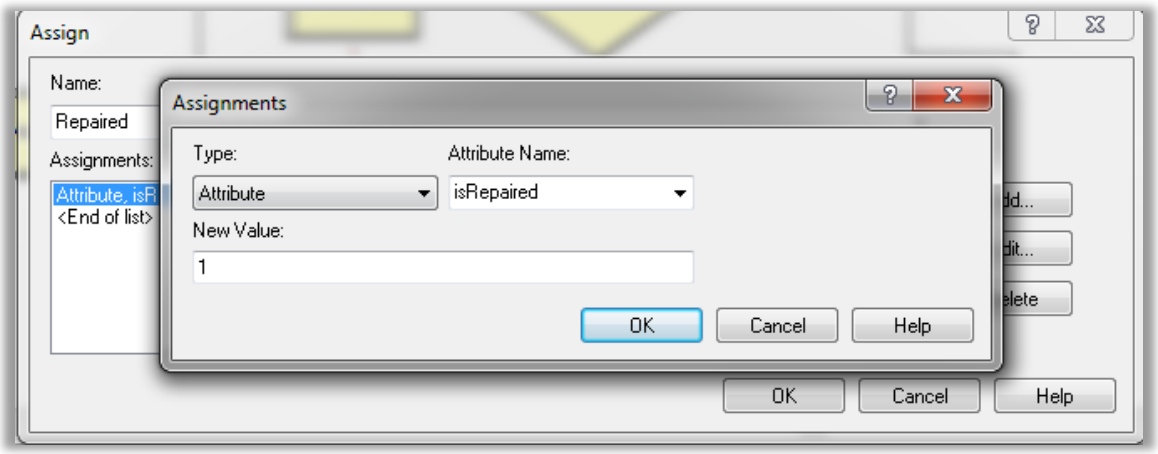


La durée de Test est définie par une Loi Exponentielle de paramètre « dureeTest ». C'est comme ça que nous utilisons l'attribut créé précédemment. Nous réitérons la manip' avec les bloc « Lavage » et « Reparation » :

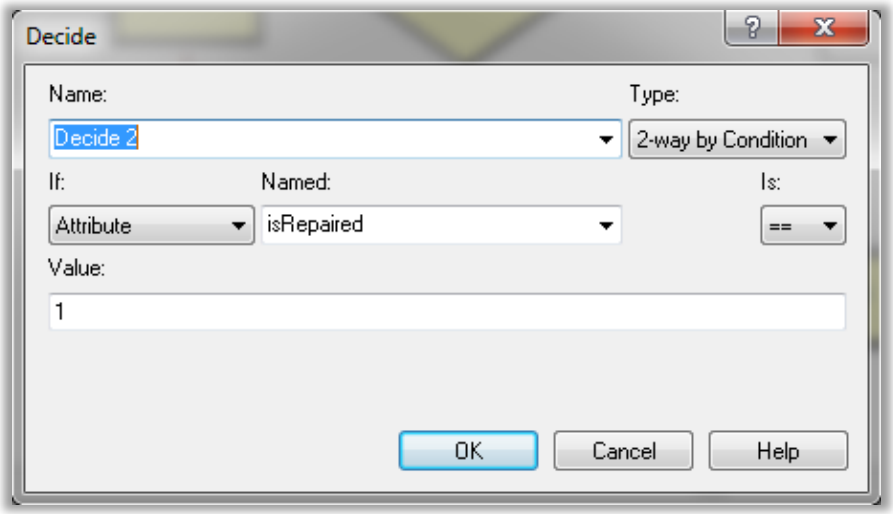




Lorsqu'un véhicule passe par la zone de réparation, nous modifions son attribut « isRepaired » à 1 en modifiant le bloc « assign » qui suit le bloc « Reparation »:

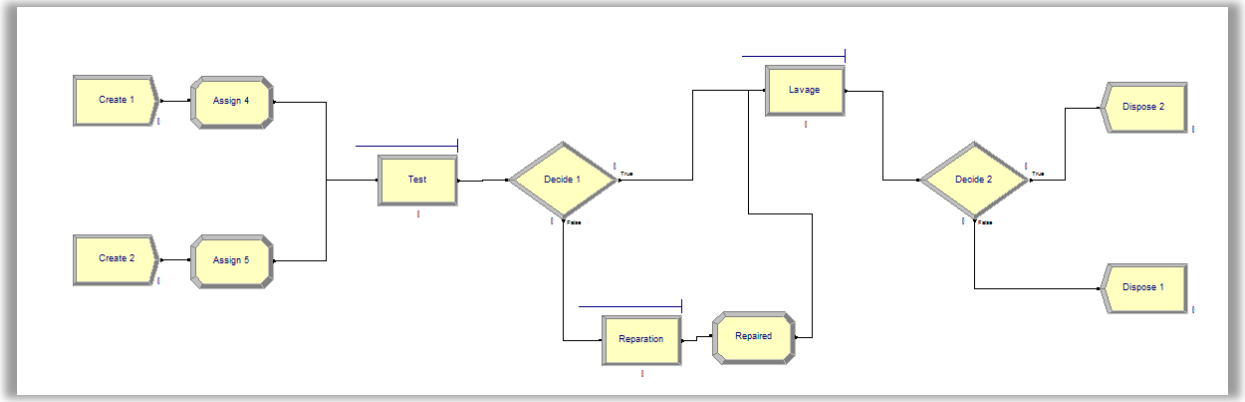


Il ne reste que 2 blocs : les blocs « Decide ». Le premier est un bloc « 2-way by Chance » classique, en fixant la probabilité à 80%. Le deuxième Decide repose sur l'attribut « isRepaired » et est défini comme suit :

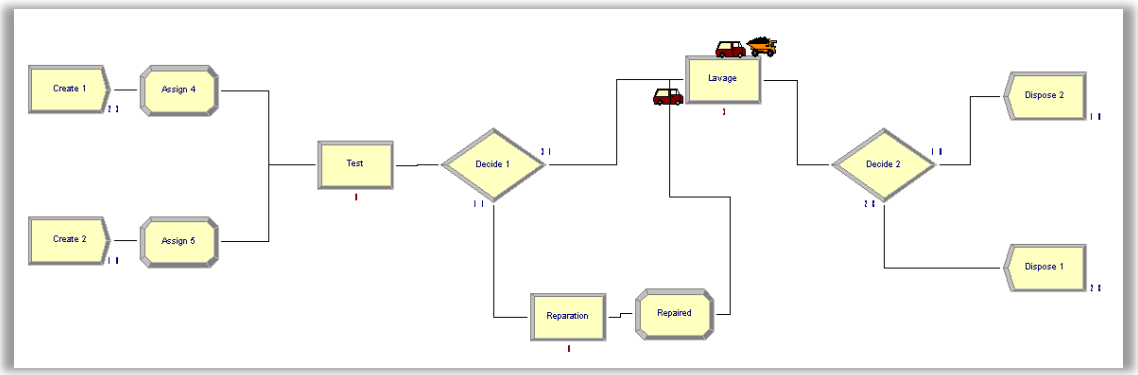


Nous modifions le type à « 2-way by Condition ». Le If repose sur un attribut et le menu déroulant « Is » prend la valeur « == ».

Pour finir il suffit de joindre tous les blocs entre eux, ce qui donne le graphique ci-dessous :



4.6. Résultat d'exécution


Create 1
Assign 4
Create 2
Assign 5
Test
Decide 1
Lavage
Reparation
Repaired
Decide 2
Dispose 2
Dispose 1