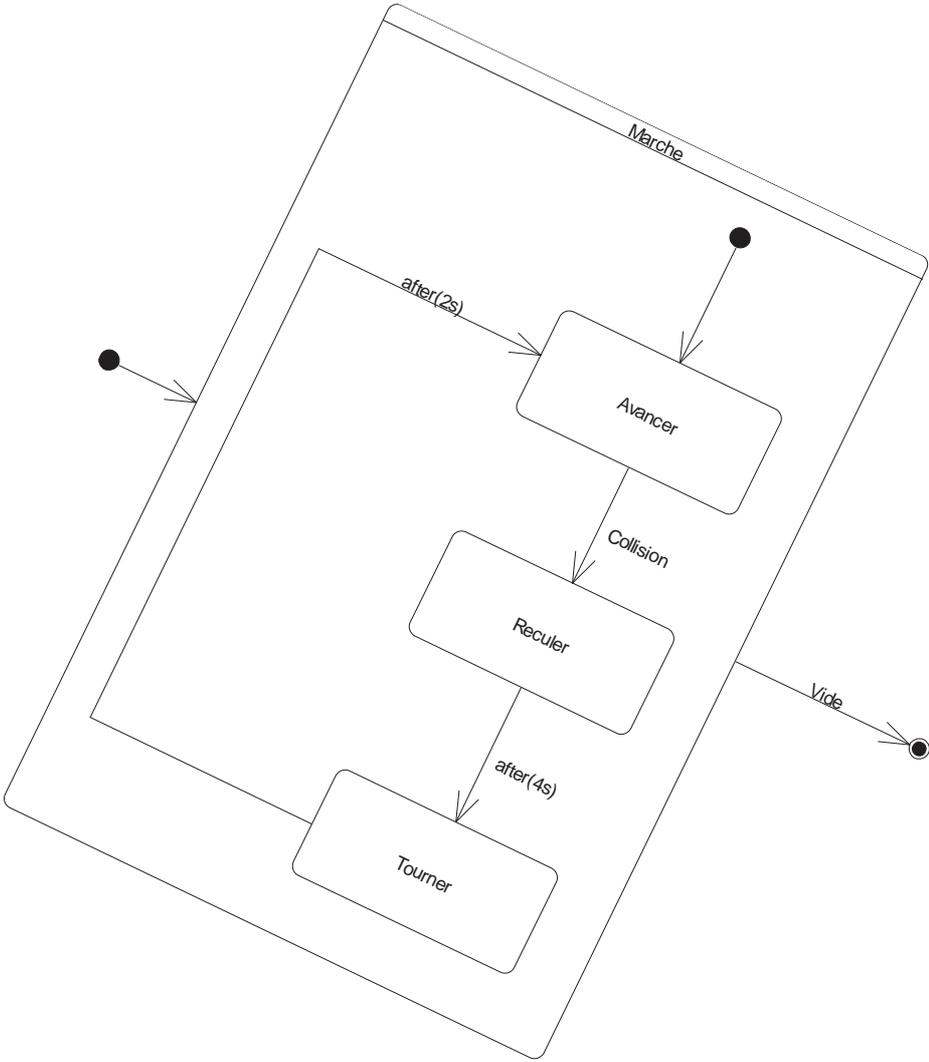


UML/SysML dans AUTOMGEN

©2010-2011 IRAI



Sommaire

Généralités	5
Editeur UML/SysML	5
Création des programmes.....	6
Représentation	6
Les éléments.....	6
Etat du programme pendant l'exécution	10
Exemple de programme.....	11

Généralités

Ce manuel a pour but d'expliquer l'utilisation d'UML et SysML intégré à AUTOMGEN. Ceci permet la création de programmes sous forme de représentation UML/SysML dans le but de générer du code. Ceci est utilisable à partir de la version 8.019 d'AUTOMGEN. Les droits pour le pack UML/SysML doivent être présents sur la licence pour pouvoir utiliser cette fonctionnalité. La première implémentation permet d'utiliser les diagrammes d'activités. Les autres formes de diagrammes seront ajoutées progressivement.

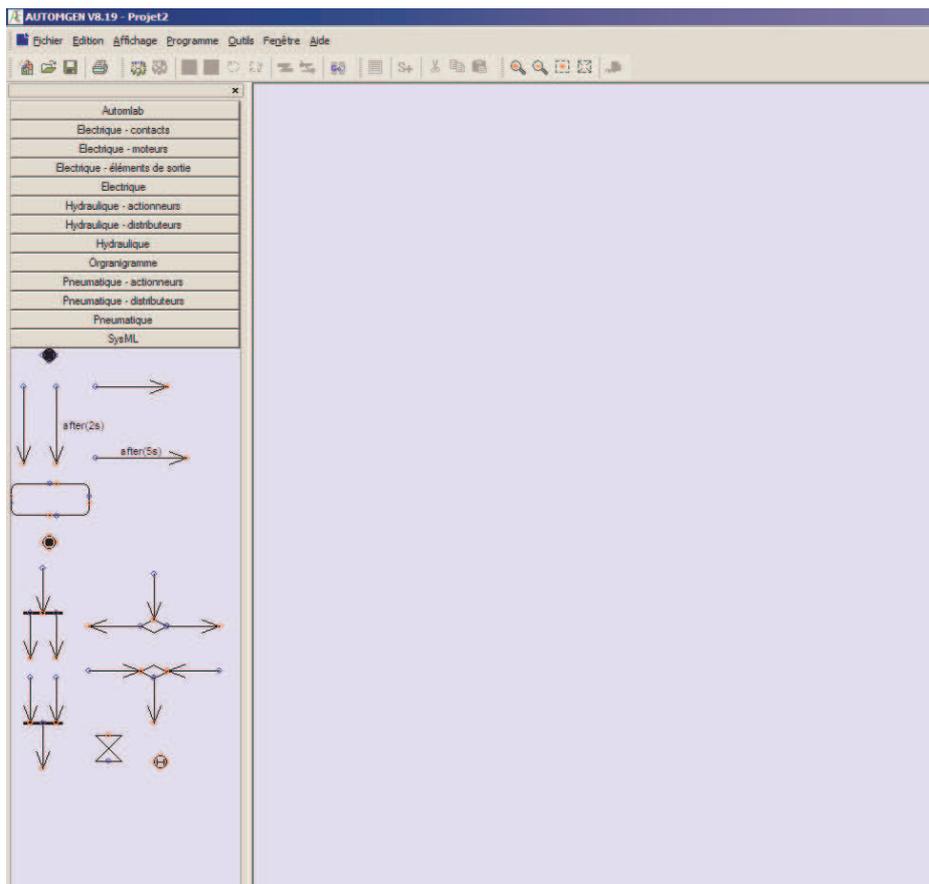
UML/SysML est vu comme un langage supplémentaire dans AUTOMGEN :

- l'ensemble des post-processeurs est utilisable,
- l'ensemble des fonctionnalités d'AUTOMGEN reste utilisable en complément des folios utilisant UML/SysML.

Editeur UML/SysML

La création des programmes est réalisé sur les folios AUTOMSIM d'AUTOMGEN.

L'éditeur se présente sous la forme suivante :



Création des programmes.

Saisissez les éléments sur la partie gauche de l'écran et déplacez-les sur la page. Vous pouvez ouvrir un assistant en cliquant avec le bouton droit de la souris sur la page et en choisissant « Ajouter un objet ».

Pour documenter les éléments, double cliquez dessus (ou sélectionnez-les, puis cliquez dessus avec le bouton droit de la souris et choisissez « Propriétés »), la boîte de dialogue des propriétés de l'objet s'ouvre alors.

Les éléments peuvent être déplacés et redimensionnés (saisir un des carrés affichés autour des éléments lorsqu'ils sont sélectionnés). Des pastilles colorées montrent les connexions des éléments. Les éléments peuvent être placés pour que les pastilles coïncident, des liaisons peuvent également être créées en cliquant sur les pastilles (le curseur affiche une forme de cible) puis en cliquant sur le chemin où doit être tracé la liaison et en finissant par un clic sur la pastille de destination. Seules les pastilles de mêmes couleurs peuvent être reliées.

Représentation

La représentation utilisée est celle d'UML/SysML. Cette notice ne détaille pas les règles d'utilisation de la norme UML/SysML, merci de vous reporter à la norme où à des manuels spécialisés si vous souhaitez apprendre les concepts d'UML/SysML.

Les éléments



Cet élément marque le début du programme. Il devrait y avoir au moins un élément « Etat initial » dans un programme. Cet élément est actif au démarrage du programme

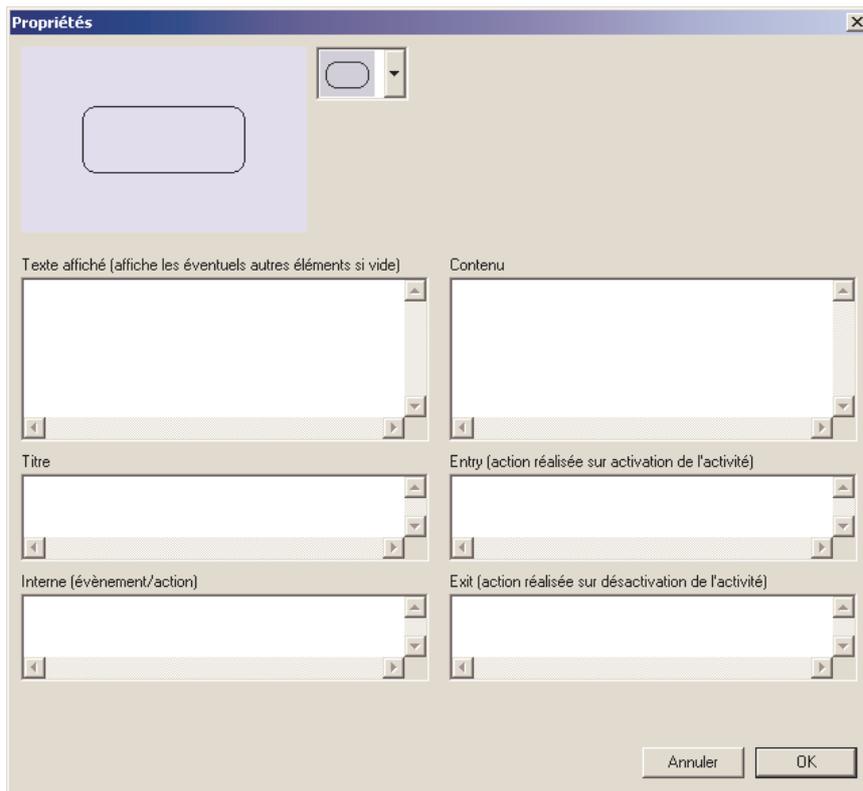


Cet élément marque la fin du programme. Cet élément peut ne pas être utilisé si le programme ne doit jamais se terminer.



Cet élément est une activité.

Voici la boîte de dialogue des propriétés d'une activité :



La zone « contenu » peut contenir du code à exécuter.

La zone « Entry » peut contenir du code qui sera exécuté une fois sur activation de l'activité.

La zone « Exit » peut contenir du code qui sera exécuté une fois sur désactivation de l'activité.

La zone « Interne » peut contenir du code qui sera exécuté de façon conditionnelle.

La syntaxe à utiliser pour ces éléments est celle du langage littéral ST décrite dans le manuel d'AUTOMGEN8.

Pour répondre à la norme UML SysML, la syntaxe « do/action » peut être utilisée.

La zone « Texte affiché » peut contenir un texte qui sera affiché à la place du code.

La zone « Titre » peut contenir un texte qui sera affiché comme titre de l'activité.

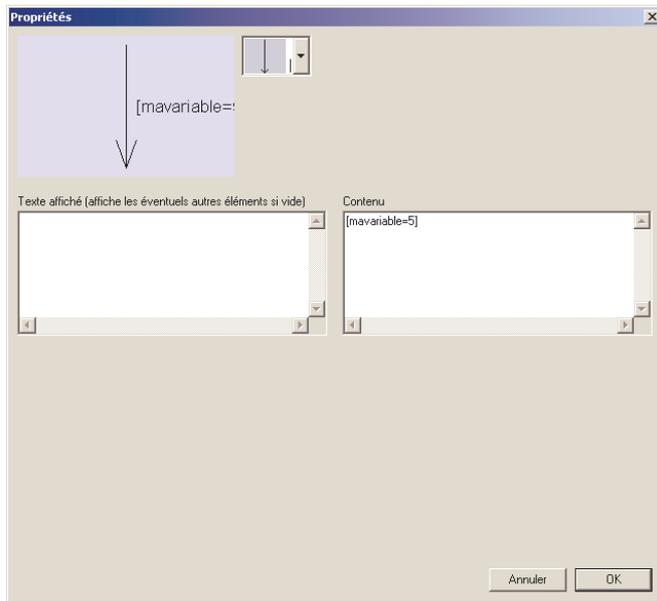


Cet élément est une transition. La syntaxe à utiliser est celle d'un test du langage ST : une expression telle qu'on peut l'écrire dans un test IF du langage littéral ST. Les tests de fronts sont réalisés avec les opérateurs RE et FE. Par exemple :

RE(%i5) (* front montant de l'entrée 5 *)

FE(%i8) (* front descendant sur l'entrée 8 *)

/%i2 (* complément de l'entrée 2 *)

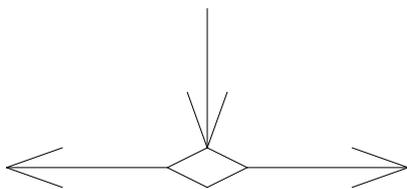


Pour répondre à la norme à la norme UML SysML :

- des crochets peuvent être utilisés pour encadrer le test,
- le test peut être encadré par « when(...) »,
- le test peut être « after(durée) », par exemple : « after(10s) ».

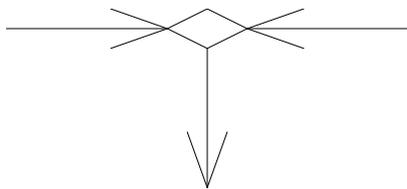
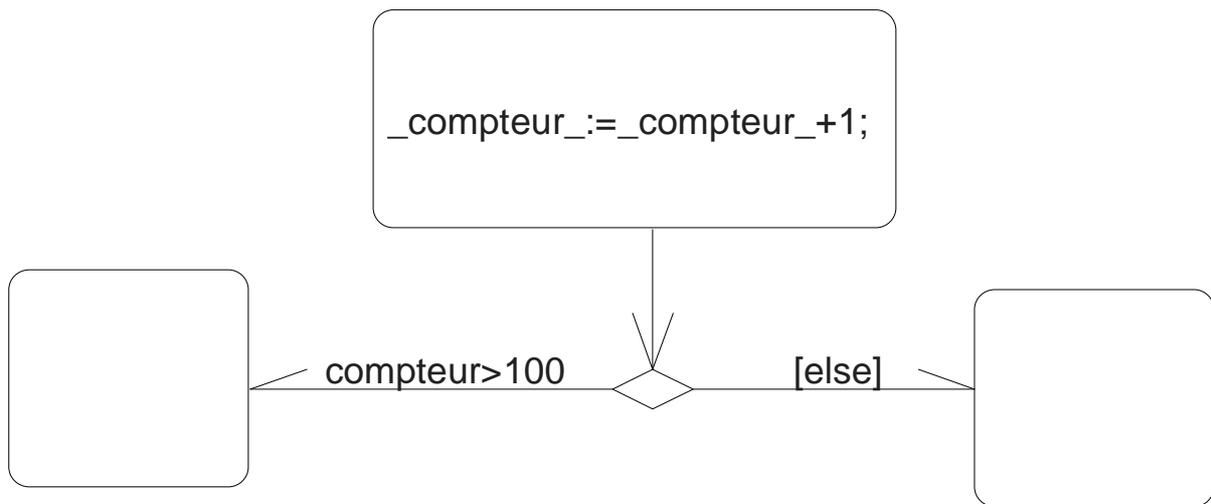
Si la zone « Contenu » est vide, la transition est toujours passante.

Ici aussi la zone texte permet de substituer un texte pour l'affichage sur la page

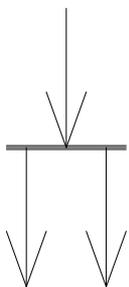


Cet élément représente une décision. Les transitions horizontales permettent de spécifier les tests. Le contenu d'une des 2 transitions peut être documenté avec le mot clé « [else] ».

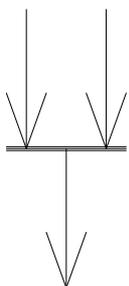
Exemple :



Cet élément est une fusion, les flux de contrôle provenant des deux transitions horizontales fusionnent.



Cet élément est une ouverture de synchronisation, le flux de contrôle provenant de la transition du haut est communiqué simultanément aux deux transitions du bas.



Cet élément est une fermeture de synchronisation. Le flux de contrôle doit arriver sur les deux transitions du haut pour être transmis à la transition du bas.



Cet élément représente une attente. La durée de l'attente est à définir dans ses propriétés.

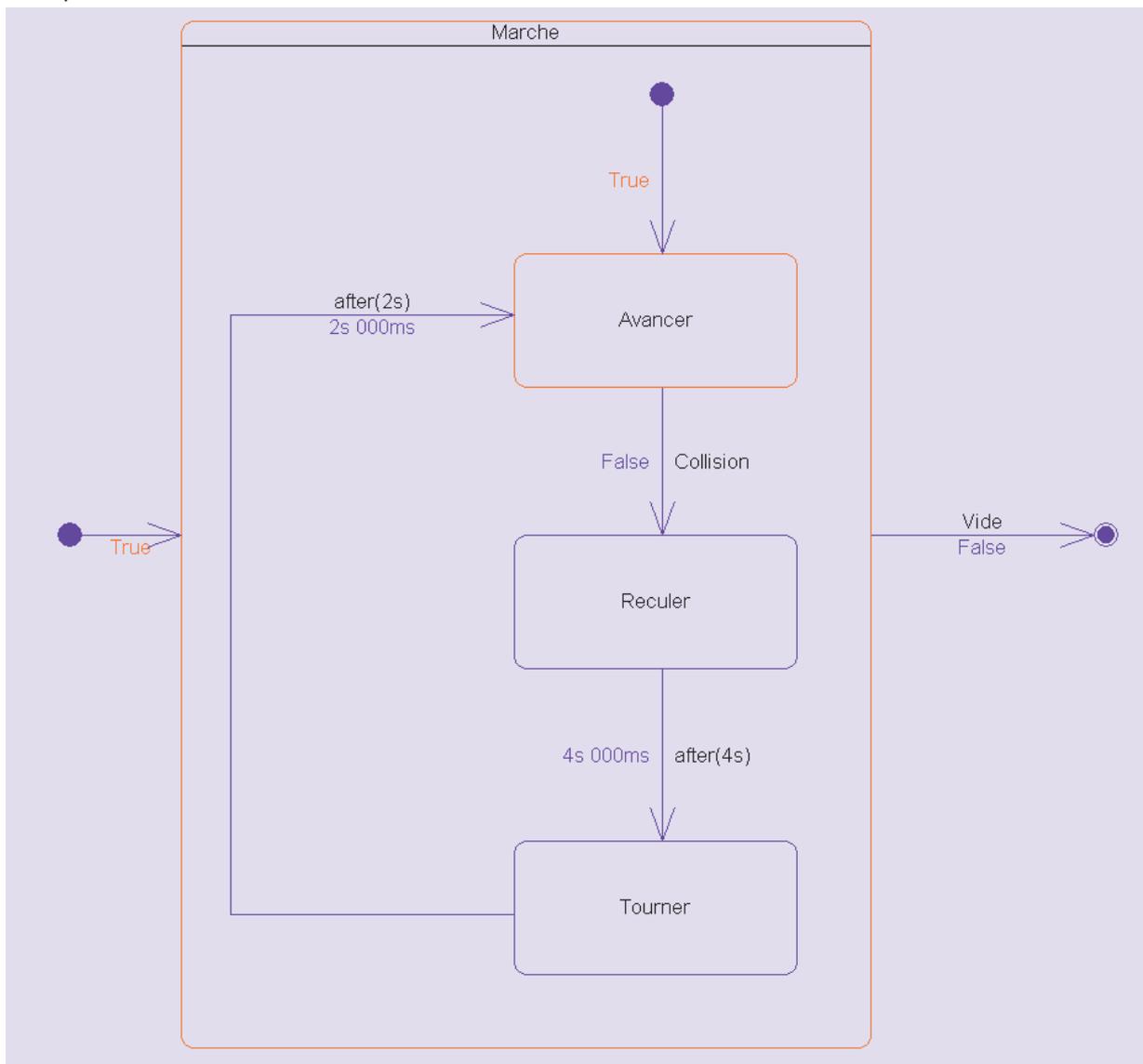


Cet élément représente l'état History (voir la norme UML SysML pour plus d'informations). Exécution

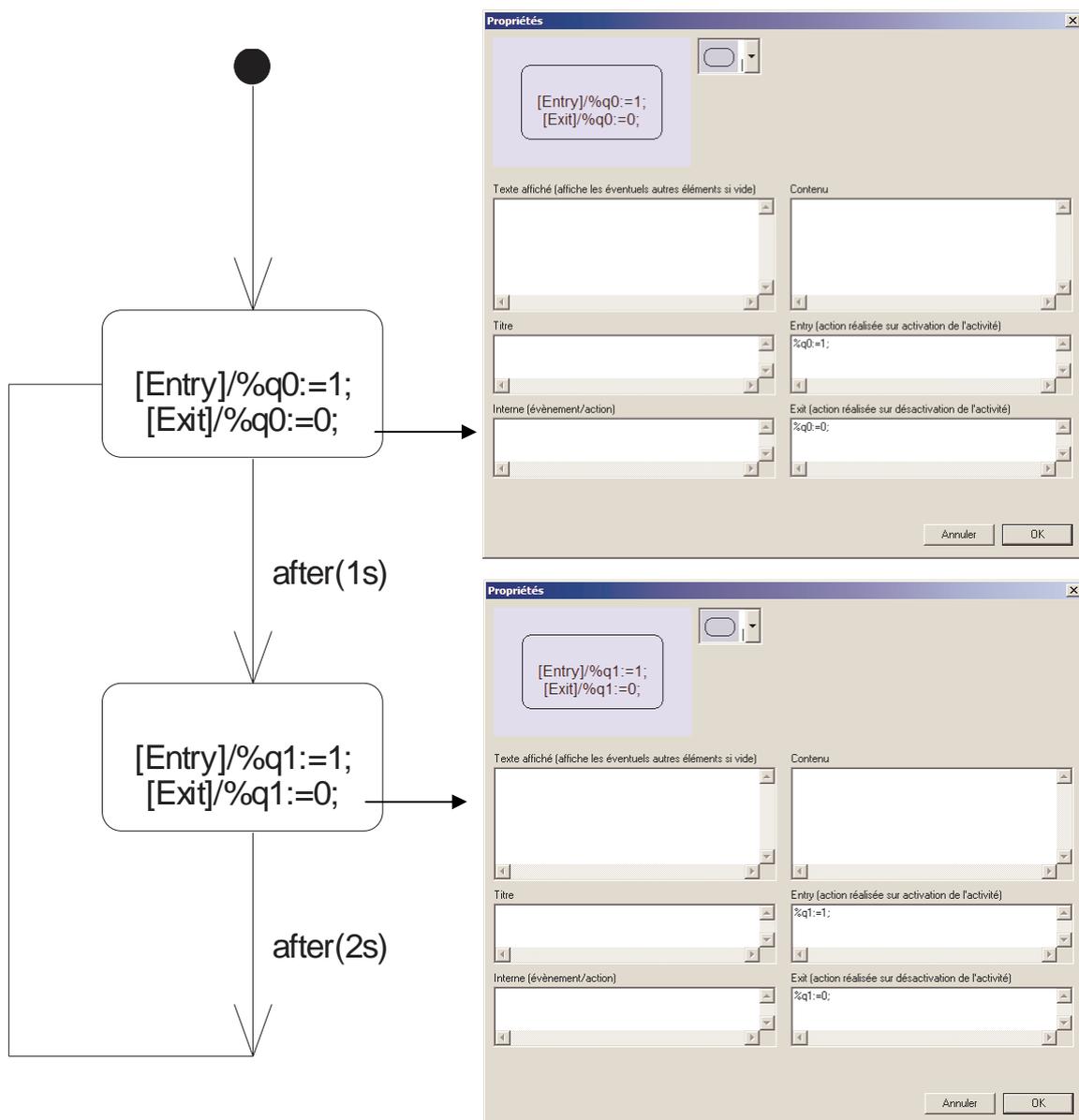
Etat du programme pendant l'exécution

Pendant l'exécution, les états des éléments sont affichés (éléments actifs en rouge, indication False/True pour les transitions) en dynamique dans l'éditeur. Les durées en cours associées aux tests liés au temps sont également affichées.

Exemple :



Exemple de programme



Ce programme réalise un clignotant, il peut, par exemple, être compilé et téléchargé sur un automate programmable.