



# Lot 1

## Architecture, Intégration

### Editeur de Systèmes de la plate-forme Calife v3.0

**Description :** Ce document présente l'interface utilisateur permettant la spécification de systèmes construits sous la forme de produits synchronisés d'automates. Les documents XML générés par cette interface suivent la structure définie la fourniture 1.1.

**Auteur(s) :** Bertrand TAVERNIER

**Référence :** AVERROES / Lot 1 / Fourniture 1.2 / V0.1

**Statut :** Non Validé

**Version :** 0.1

## **Historique**

23 Octobre 2003	V 0.1	Version Préliminaire
-----------------	-------	----------------------

# SOMMAIRE

<b>1</b>	<b>CONSTRUCTION D’UN PROJET</b> .....	<b>6</b>
1.1	CREATION D’UN NOUVEAU PROJET .....	6
1.2	OUVERTURE D’UN PROJET EXISTANT .....	6
1.3	PANNEAU DE VISUALISATION DU CONTENU D’UN PROJET .....	7
<b>2</b>	<b>CREATION D’UN COMPOSANT</b> .....	<b>9</b>
<b>3</b>	<b>EDITION D’UN COMPOSANT</b> .....	<b>10</b>
3.1	EDITION DE L’ENVIRONNEMENT.....	10
3.1.1	<i>Création d’une variable (resp. paramètre)</i> .....	11
3.1.2	<i>Modification d’une variable existante</i> .....	12
3.1.3	<i>Suppression d’une variable existante</i> .....	12
3.2	CONSTRUCTION D’UN AUTOMATE.....	12
3.2.1	<i>Création et modification d’une localité</i> .....	13
3.2.2	<i>Création et modification d’un déplacement (transition d’action)</i> .....	14
3.3	CONSTRUCTION D’UN BLOCK.....	15
3.3.1	<i>Spécification des composants</i> .....	15
3.3.2	<i>Edition des liens</i> .....	16
3.3.3	<i>Edition de la table de synchronisation</i> .....	16
3.3.4	<i>Tables de synchronisation automatiques</i> .....	17

## TABLE DES FIGURES

Figure 1 : Vue d'ensemble de la plate-forme Calife.....	5
Figure 2 : Création d'un nouveau projet.....	6
Figure 3 : Ouverture d'un projet existant.....	7
Figure 4 : Visualisation d'un projet.....	7
Figure 5 : Création d'un nouveau composant.....	9
Figure 6 : Edition d'un composant .....	10
Figure 7 : Visualisation de l'environnement .....	11
Figure 8 : Création d'une nouvelle variable .....	11
Figure 9 : Grapheur associé au composant Sender .....	12
Figure 10 : Création d'une localité .....	13
Figure 11 : Exemple de localités (avec prédicat Activité) .....	14
Figure 12 : Création d'un déplacement.....	14
Figure 13 : Panneau de construction d'un block.....	15
Figure 14 : Edition de la table de synchronisation.....	17
Figure 15 : Edition d'une composante du vecteur .....	17
Figure 16 : Assistant de création des synchronisations .....	17

# INTRODUCTION

La plate-forme Calife est constituée de 3 modules principaux :

- Un éditeur de système permettant de modéliser, de manière graphique, un système sous la forme de produits synchronisés d'automates. Le formalisme utilisé est celui des automates génériques présentés au paragraphe suivant.
- Un compilateur de modèle permettant de vérifier la bonne formation des contraintes (gardes, invariants, ...) en regard du modèle sur lequel est construit le système (automates temporisés, hybrides,...). Le compilateur de modèle vérifie également la bonne déclaration des variables, des paramètres,...
- Un moteur de scripts permettant l'exécution de scripts associés à l'exportation des modèles vers les différents outils interfacés par la plate-forme.

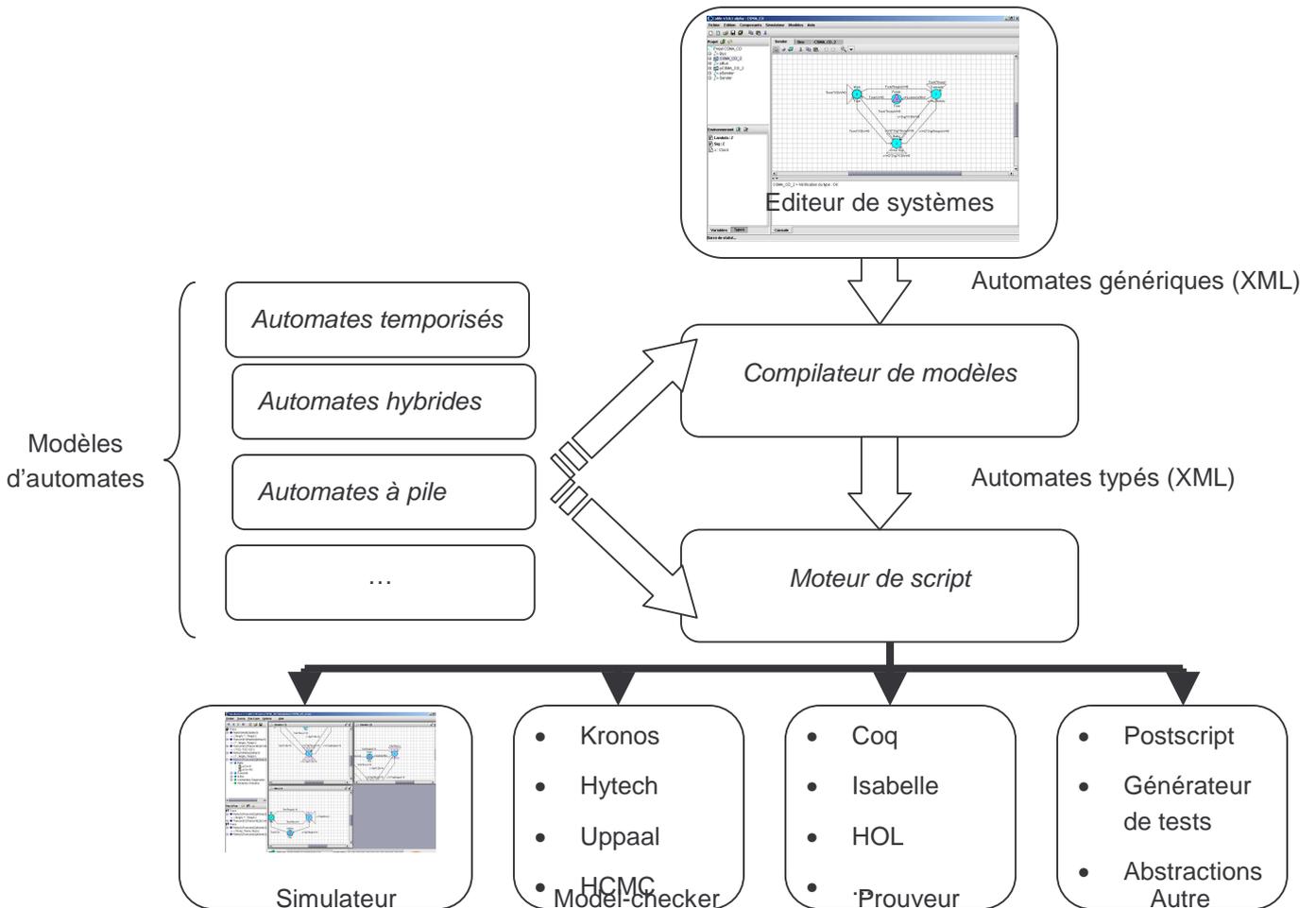


Figure 1 : Vue d'ensemble de la plate-forme Calife

Le présent document décrit le module « Editeur de Systèmes ». Son style est volontairement orienté « Manuel utilisateur ».

Par la suite, ce document a pour objectif d'être intégré au manuel de la plate-forme Calife.

# 1 CONSTRUCTION D'UN PROJET

Un projet, au sein de l'environnement Calife, est constitué d'un sous-répertoire du répertoire  $\{\text{WorkDir}\}$ . Ce dernier, racine de l'ensemble des projets Calife, est défini au moment de l'installation du logiciel. Il peut être modifié à tout moment, en éditant le fichier « .calife » situé dans le répertoire « HOME » de l'utilisateur.

Un répertoire « Projet » contiendra l'ensemble des fichiers « .clf » définissant les différents composants du projet. Par la suite, ce répertoire contiendra également tous les fichiers créés lors de l'exportation d'un composant.

*Remarque :* le répertoire  $\{\text{WorkDir}\}$  contient impérativement le fichier « Component.dtd ». Si ce fichier est absent, il est automatiquement recréé au démarrage du logiciel.

## 1.1 Création d'un nouveau projet

---

La création d'un nouveau projet est réalisée par l'intermédiaire du menu « **Fichier à Nouveau Projet...** » ou par l'icône .

Une fenêtre de dialogue apparaît pour saisir le nom du nouveau projet. Cette fenêtre indique également le répertoire associé au projet créé :

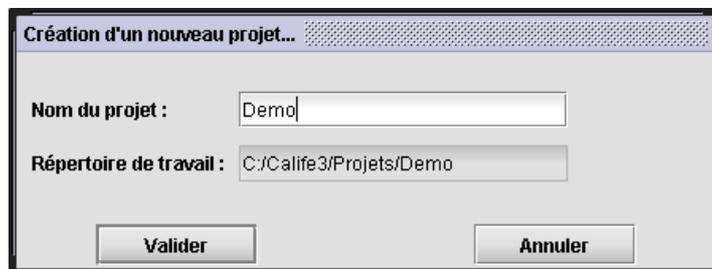


Figure 2 : Création d'un nouveau projet

Lorsque le nom du répertoire saisi est valide, le répertoire Projet est créé et automatiquement ouvert par l'éditeur de systèmes.

## 1.2 Ouverture d'un projet existant

---

L'ouverture d'un projet existant est réalisée par l'intermédiaire du menu « **Fichier à Ouvrir Projet...** » ou par l'icône .

Une fenêtre de dialogue apparaît alors pour saisir le nom du répertoire Projet :

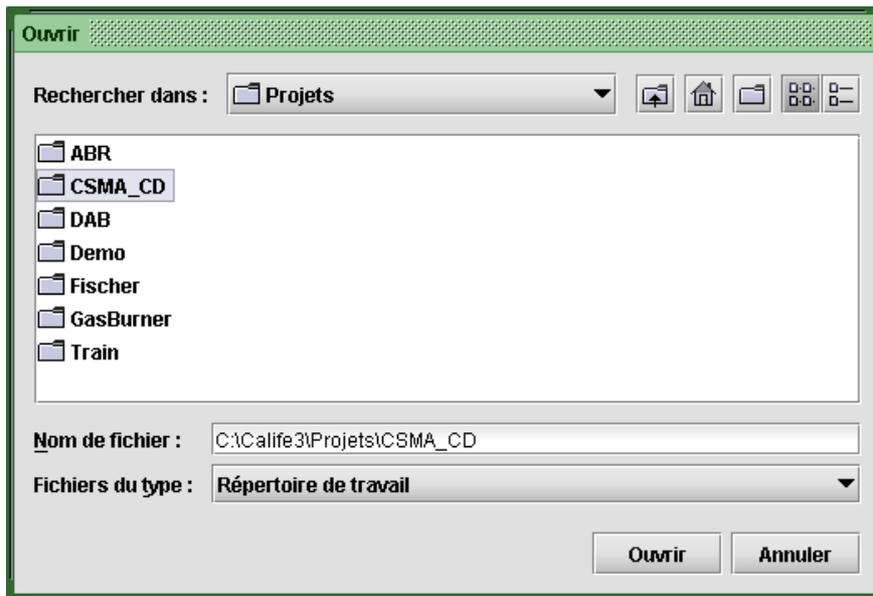


Figure 3 : Ouverture d'un projet existant

La sélection du projet à ouvrir est réalisée en cliquant sur le nom du projet depuis la liste proposée puis sur le bouton « **Ouvrir** ».

### 1.3 Panneau de visualisation du contenu d'un projet

Ce panneau est situé en haut, à gauche de l'éditeur :

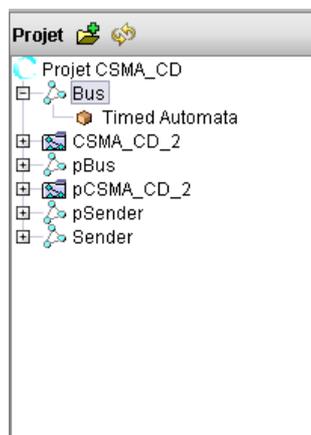


Figure 4 : Visualisation d'un projet

Il permet de visualiser l'ensemble des composants du projet (associés à des fichiers .clf situés dans le répertoire projet).

Le panneau projet permet :

- De sélectionner à l'aide d'un double-click sur son nom, un composant à éditer.
- D'insérer, à l'aide d'une action de type glisser-déplacer, un composant dans un block
- De recopier un composant existant d'un autre projet vers le projet actif (bouton )

- De rafraîchir le contenu de l'arborescence Projet (bouton ) pour vérifier si de nouveaux composants ont été insérés dans le répertoire projet.
- De vérifier le typage d'un composant à la vue du système de type défini dans le modèle. Cette action est réalisée à l'aide du menu contextuel apparaissant en cliquant à l'aide du bouton droit sur le nom d'un composant.
- D'exporter un composant. Cette action est réalisée à l'aide du menu contextuel apparaissant en cliquant à l'aide du bouton droit sur le nom d'un composant.

## 2 CREATION D'UN COMPOSANT

La création d'un nouveau composant est réalisée par l'intermédiaire du menu « **Composant à Nouveau Composant ...** » ou par l'icône .

Une fenêtre de dialogue apparaît alors pour saisir :

- Le nom du nouveau composant. Ce nom détermine le nom du fichier « .clf ». Il n'est donc pas possible, au sein du même projet, de créer deux composants ayant le même nom.
- Le modèle sur lequel ce composant est construit. L'éditeur propose tous les modèles valides contenus dans le répertoire **Modeles** du répertoire d'installation de Calife v3.
- Le type de composant (Automate ou Block)

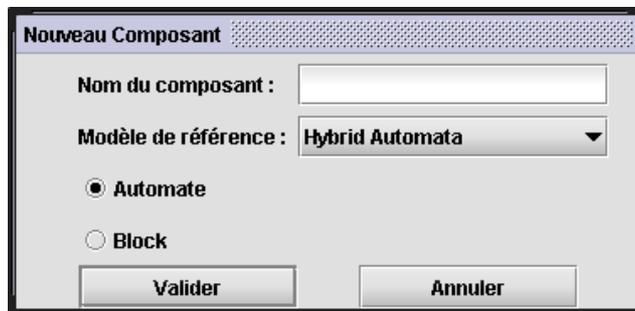


Figure 5 : Création d'un nouveau composant

Le menu « Composants » permet de réaliser des actions spécifiques à la gestion des composants. En particulier, ce menu propose les actions suivantes :

- Création d'un composant,
- Importation d'un composant existant dans le projet courant,
- Suppression d'un composant,
- Renommage d'un composant,
- Vérification du typage d'un composant,
- Changement du modèle de référence d'un composant
- Edition, lorsque le modèle le permet, de certains éléments du système de type.

### 3 EDITION D'UN COMPOSANT

D'un point de vue général, l'édition d'un composant (qu'il s'agisse d'un automate ou d'un block) est réalisée en ouvrant l'éditeur associé à celui-ci.

Cette action se fait par l'intermédiaire d'un double-click sur le nom du composant dans l'arborescence « Projet ». Un onglet dont le nom est celui du composant est alors ajouté dans la partie centrale de l'éditeur. Si cet onglet était déjà présent dans la partie centrale, il est alors sélectionné.

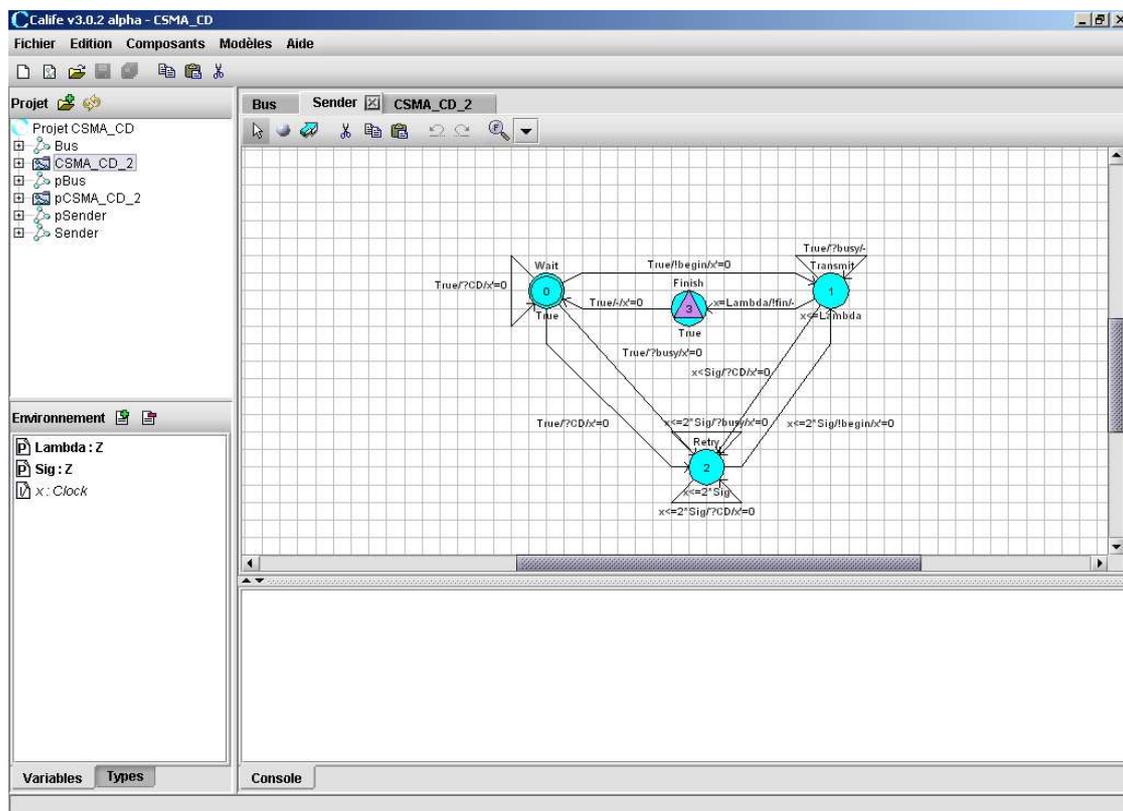


Figure 6 : Edition d'un composant

*Remarque :* les onglets associés aux éditeurs peuvent être fermés par l'intermédiaire de la croix apparaissant au passage de la souris à côté du nom du composant sur l'onglet.

#### 3.1 Edition de l'environnement

Le panneau *Environnement* situé en bas à gauche, est toujours associé au composant actif dans la partie éditeur à onglets (partie centrale).

Ce panneau permet de visualiser d'une part l'ensemble des variables et des paramètres définis pour le composant (onglet **Variables**), d'autre part le système de types associé (onglet **Types**).

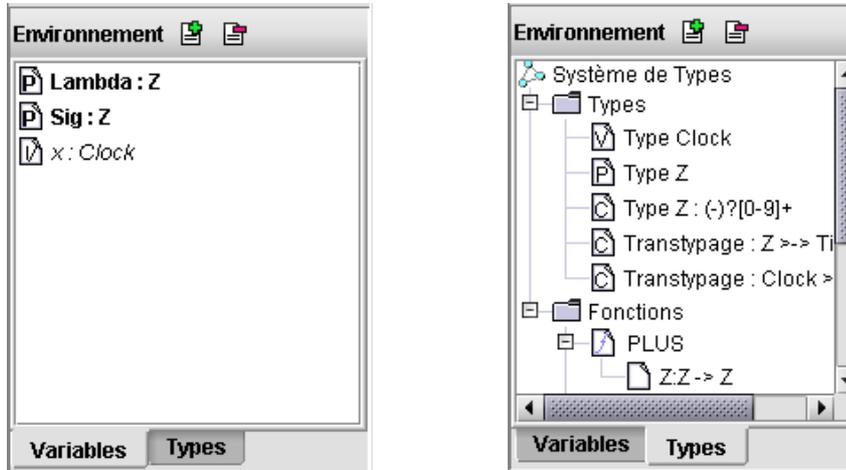


Figure 7 : Visualisation de l'environnement

L'onglet *Variables* permet la création, la modification ou la suppression de variables et de paramètres.

### 3.1.1 Création d'une variable (resp. paramètre)

La création d'une nouvelle variable est réalisée par l'intermédiaire de l'icône  située en haut du panneau *Environnement*.

Une fenêtre de dialogue, permettant la saisie du nom, du type et des attributs de la variable (resp. paramètre) s'ouvre alors :

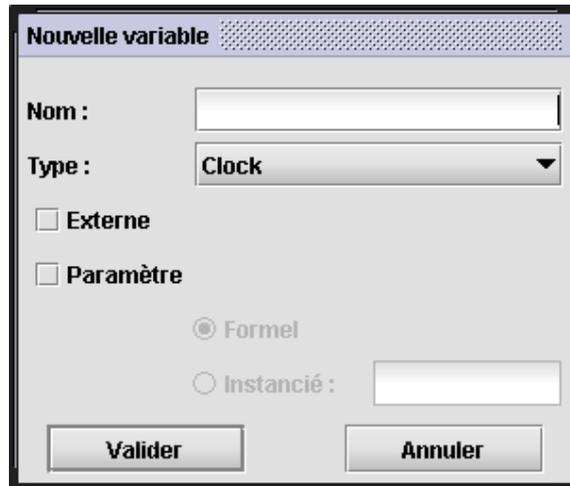


Figure 8 : Création d'une nouvelle variable

Dans cette fenêtre, la liste déroulante associée au type est différente pour une variable ou un paramètre (case *Paramètre* cochée).

La case à cocher *Externe* permet de spécifier la portée du paramètre ou de la variable (Local/Externe).

Lorsque la case *Paramètre* est cochée et que le paramètre est local (case *Externe* non cochée), il est possible d'associer une valeur au paramètre en sélectionnant *Instancié* et en saisissant la valeur dans la zone de saisie.

### 3.1.2 Modification d'une variable existante

La modification d'une variable existante est réalisée en double-cliquant sur le nom de la variable depuis la liste des variables.

Une fenêtre de dialogue, identique à la précédente est alors ouverte.

### 3.1.3 Suppression d'une variable existante

La suppression d'une variable (resp. un paramètre) existante est réalisée en sélectionnant la variable dans la liste et en cliquant sur l'icône .

## 3.2 Construction d'un automate

La construction d'un automate est réalisée de manière graphique, par l'intermédiaire d'un grapheur associé au composant :

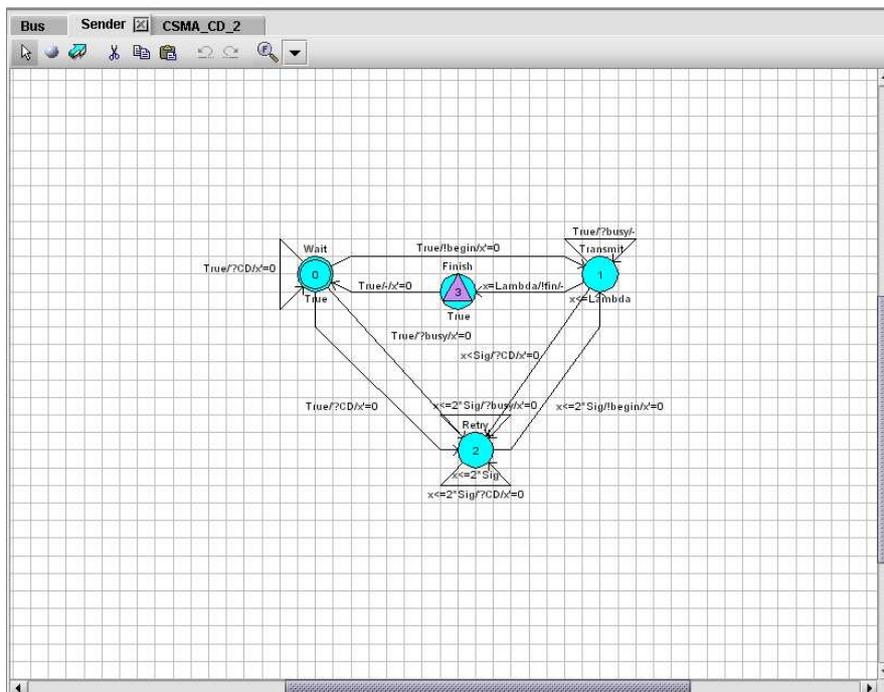


Figure 9 : Grapheur associé au composant Sender

Le grapheur propose les fonctionnalités suivantes :

- Fonctions de Sélection. Ces fonctions sont actives en se positionnant en mode *Sélection*, par l'intermédiaire de l'icône  de la barre d'outils. Les sélections peuvent alors être réalisées :

- En cliquant sur un élément du graphe (Localité, Texte, Nœud intermédiaire d'une transition)
- En dessinant un rectangle à l'aide de la souris, par une action de type glisser-déplacer.
- En double-cliquant sur un des éléments d'une transition (Texte ou Nœud intermédiaire). Dans ce cas, toute la transition est alors sélectionnée.
- Fonctions de Zoom : 3 modes de zoom sont disponibles par l'intermédiaire de la liste déroulante à droite sur la barre d'outils. Le type de zoom sélectionné est réalisé en cliquant sur le bouton sélectionné dans la liste déroulante. Les 3 fonctionnalités de zoom sont :
  - Augmenter le zoom par l'icône 
  - Diminuer le zoom par l'icône 
  - Choisir l'échelle la plus appropriée par l'icône 
- Fonctions d'éditations avancées : les fonctionnalités sont les suivantes :
  - Copier le contenu d'une sélection par l'icône  ou *CRTL+C*.
  - Couper la sélection par l'icône  ou *CRTL+X* (si une sélection contient une transition partiellement sélectionnée, celle-ci sera alors supprimée).
  - Coller le contenu du presse-papier par l'icône  ou *CRTL+V*. Cette opération n'est possible que si une sélection a été préalablement copiée ou coupée.
  - Défaire des modifications par l'icône 
  - Refaire des modifications défaites par l'icône 

### 3.2.1 Création et modification d'une localité.

La création d'une localité est réalisée en se plaçant au préalable dans le mode « Insertion de Localité », par l'intermédiaire de l'icône .

Il suffit alors de cliquer sur une zone libre du graphe pour faire apparaître la fenêtre de dialogue suivante :

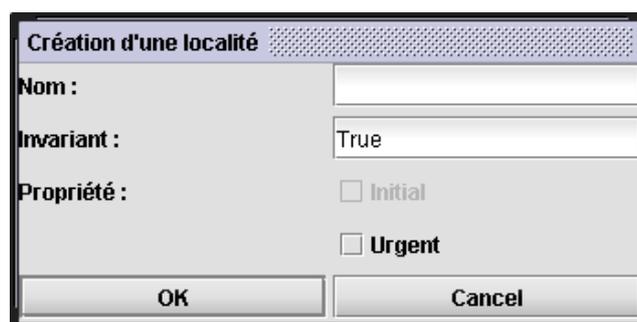


Figure 10 : Création d'une localité

Il est alors possible de :

- spécifier un nom pour la localité. Un nom du type e<numéro> sera défini par défaut .
- saisir un prédicat associé à l'invariant. La grammaire du prédicat dépend du système de type défini dans le modèle. Néanmoins, certains opérateurs sont pretty-printés au niveau du parseur et automatiquement traduits sous la forme de fonctions associées (définies ou non...). Cette

remarque concerne les fonctions suivantes (définies dans tous les modèles fournis avec la plate-forme) :

- Fonctions « AND » pretty-printé « && »
- Fonction « OR » pretty-printé « || »
- Fonctions de comparaison « LT », « LTE », « GT », «GTE », « EQ », « NEQ » respectivement pretty-printés « < », « <= », « > », « >= », « = », « != ».
- Fonctions usuelles « PLUS », « MOINS », « MULT » pretty-printées « + », « - », « \* »
- saisir un prédicat associé à l'activité dans la localité (optionnel). Dans ce cas, le prédicat associé à l'invariant sera suivi du symbole « ; » puis du prédicat associé à l'activité. Par exemple, l'automate « gas burner » (construit sur le modèle des automates hybrides, fournit avec la plate-forme) utilise ce genre de constructions :

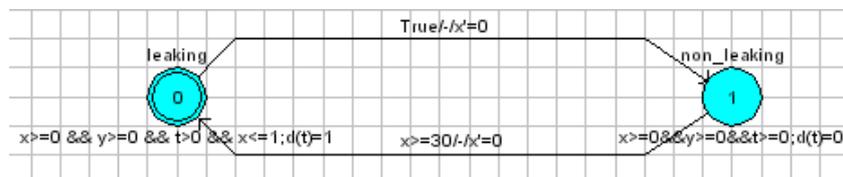


Figure 11 : Exemple de localités (avec prédicat Activité)

- saisir certaines propriétés liées au nœud (état initial, état urgent).

*Remarque :* il n'est possible de sélectionner qu'un état initial. Cette sélection est donc impossible lorsque un état initial est déjà défini dans l'automate.

La localité ainsi créée peut alors être déplacée sur le graphe en la sélectionnant puis en réalisant une action de type glisser-déplacer.

La modification des informations associées à une localité existante peut-être réalisée en sélectionnant la localité (par un simple ou double-click) et en cliquant à l'aide du bouton droit.

Une fenêtre de dialogue, identique à celle décrite ci-dessus s'ouvre alors.

### 3.2.2 Création et modification d'un déplacement (transition d'action).

La création d'un déplacement est réalisée en passant au préalable dans le mode « Insertion de déplacements », par l'intermédiaire de l'icône

Il suffit alors de cliquer sur la localité de départ du déplacement. En déplaçant la souris, il est possible de visualiser l'arc en cours de création. On peut alors insérer des nœuds intermédiaires pour fragmenter l'arc. Le fait de cliquer sur une localité d'arrivée entraîne l'ouverture de la fenêtre de dialogue suivante :

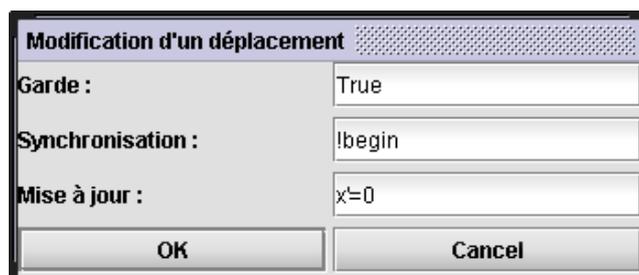


Figure 12 : Création d'un déplacement

Il est alors possible de saisir :

- Le prédicat associé à la garde
- Un label utilisé par la suite pour synchroniser le composant (optionnel)
- Une relation de mise à jour sous la forme de prédicats séparés par le symbole « ; »

Remarque : les labels de synchronisation peuvent commencer par le symbole « ! » ou « ? ». Cette information n'a pas de sémantique précise, mais sera utilisée par certains algorithmes de création automatique de la table de synchronisation (cf. paragraphe 3.3.3).

La modification des informations associées à un déplacement existant peut-être réalisée en sélectionnant le déplacement complet (par double-click) et en cliquant à l'aide du bouton droit.

Une boîte de dialogue, identique à celle décrite ci-dessus s'ouvre alors.

### 3.3 Construction d'un block

La construction d'un block (produit synchronisé) est réalisée par l'intermédiaire d'un panneau de saisie des informations nécessaires (composants, liens et table de synchronisation) :

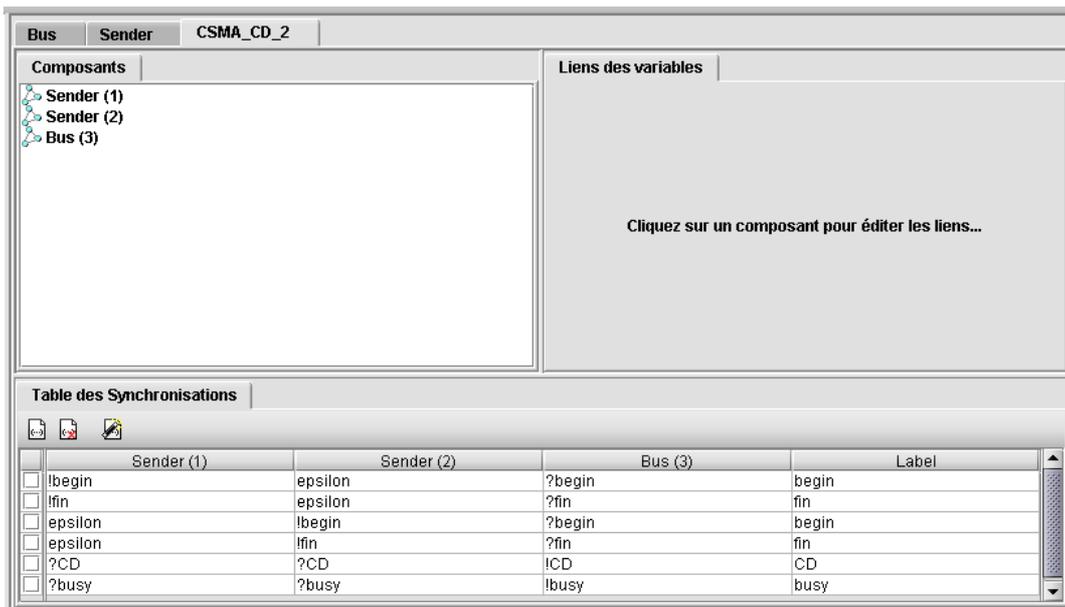


Figure 13 : Panneau de construction d'un block

#### 3.3.1 Spécification des composants

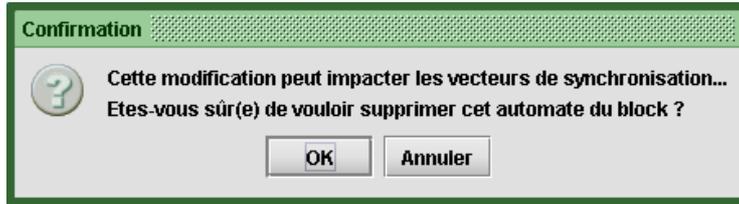
Les composants constituant le block sont visibles dans la liste présente sous l'onglet « Composants ».

Il est possible d'ajouter des composants dans cette liste en réalisant une action de type glisser-déplacer, depuis un composant de l'arborescence *Projet* vers cette liste.

Un index numérique est alors automatiquement attribué au composant et une colonne est ajoutée à la table de synchronisation.

Si celle-ci contient déjà des vecteurs de synchronisation, les composantes ajoutées sont positionnées sur le label epsilon.

Il est possible de supprimer un composant de la liste en le sélectionnant et en appuyant sur la touche *Suppr.* du clavier. Un message de confirmation apparaît alors :

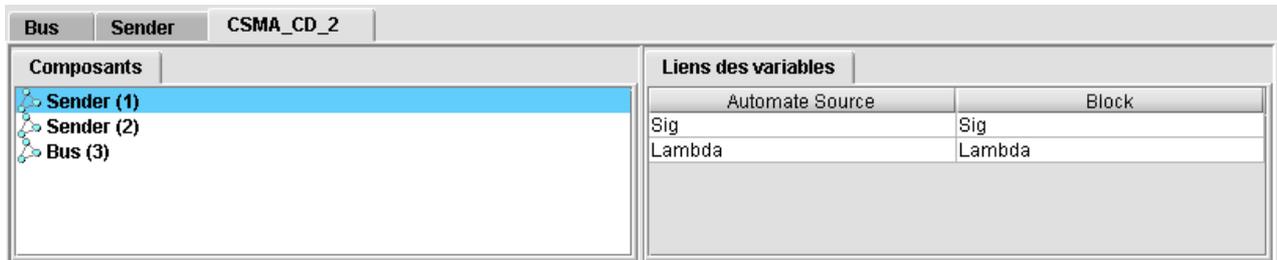


Cliquez sur OK pour continuer ou Annulez.

Remarque : la colonne de la table de synchronisation associée au composant supprimé sera également supprimée en cas de confirmation.

### 3.3.2 Edition des liens

Les liens spécifiques à un composant apparaissent lorsque celui-ci est sélectionné depuis la liste « Composants » :



Le panneau « Liens des variables » permet de visualiser à gauche les variables « externes » du composant sélectionné et à droite la variable du block liée à la variable de l'automate.

La variable du block peut être modifiée en cliquant sur la cellule. Une liste déroulante propose alors toutes les variables du block. Il est également possible de choisir « -- Valeur Constante -- » dans cette liste. Une boîte de dialogue permet alors de saisir la valeur choisie.

Remarque : A l'insertion d'un nouveau composant dans le block, l'éditeur réalise alors un lien automatique entre les variables et les paramètres portant le même nom.

### 3.3.3 Edition de la table de synchronisation

L'éditeur associé à la table de synchronisation se présente sous la forme d'un tableau. Chaque colonne est associée à un composant, à l'exception de la dernière colonne utilisée pour saisir le nom du vecteur de synchronisation.

Chaque ligne de ce tableau peut donc être lue comme un vecteur de synchronisation.

Table des Synchronisations				
	Sender (1)	Sender (2)	Bus (3)	Label
<input type="checkbox"/>	!begin	epsilon	?begin	begin
<input type="checkbox"/>	!fin	epsilon	?fin	fin
<input type="checkbox"/>	epsilon	!begin	?begin	begin
<input type="checkbox"/>	epsilon	!fin	?fin	fin
<input type="checkbox"/>	?CD	?CD	!CD	CD
<input type="checkbox"/>	?busy	?busy	!busy	busy

**Figure 14 : Edition de la table de synchronisation**

La barre d'outils située en haut du panneau permet :

- d'insérer un nouveau vecteur de synchronisation à l'aide de l'icône 
- de supprimer les vecteurs sélectionnés à l'aide de la case à cocher (colonne la plus à gauche), à l'aide de l'icône 
- d'appeler l'assistant de création automatique de tables de synchronisation à l'aide de l'icône 

Lorsqu'un nouveau vecteur de synchronisation est créé, celui-ci est initialisé avec toutes ses composantes à la valeur « epsilon » et « Action » comme nom.

Cliquer sur une composante du vecteur permet de faire apparaître une liste déroulante contenant l'ensemble des labels introduits dans les transitions de l'automate. Il est alors possible de sélectionner, pour chaque composant, les labels synchronisés entre eux.

	Sender (1)	Sender (2)	Bus (3)	Label
<input type="checkbox"/>	!begin	epsilon	?begin	begin
<input type="checkbox"/>	!fin	epsilon	?fin	fin
<input type="checkbox"/>	epsilon	!begin	?begin	begin
<input type="checkbox"/>	epsilon	!fin	?fin	fin
<input type="checkbox"/>	?CD	?CD	!CD	CD
<input type="checkbox"/>	?busy	?busy	!busy	busy
			epsilon	

**Figure 15 : Edition d'une composante du vecteur**

Le label de la synchronisation peut être modifié en saisissant dans la case le nouveau label du vecteur.

### 3.3.4 Tables de synchronisation automatiques

L'assistant de création de table de synchronisation apparaît par l'intermédiaire de l'icône  :

**Assistant de création des synchronisations**

**Rendez-vous**  
 **BroadCast**  
 **Binaire**

**Description**  
 La synchronisation est de type "Chaque émission d'un message concerne tous les récepteurs potentiels"  
 L'émetteur reste donc bloqué jusqu'à ce que tous les récepteurs soient prêt à recevoir le message.

Remplacer la table de synchronisation existante

**Figure 16 : Assistant de création des synchronisations**

L'assistant propose alors un certain nombre d'algorithmes de création de tables de synchronisation :

- La synchronisation par rendez-vous : le principe consiste à synchroniser chaque label commençant par le symbole « ! » puis une chaîne de caractères avec tous les labels des autres composants contenant un label de la forme « ? » puis la même chaîne de caractères.
- La synchronisation par broadcast, déduite de la synchronisation par rendez-vous : par opposition à la synchronisation par rendez-vous où tous les composants contenant le label ?Action sont concernés, la synchronisation par broadcast va créer tous les sous-vecteurs concernant de 0 à n composants contenant le label ?Action. Ce type de synchronisation permet de modéliser l'envoi de messages non-bloquants pour l'émetteur.
- La synchronisation binaire est une synchronisation ne concernant que 2 composants. On synchronise donc chaque label de type !Action de chaque composant avec chaque label de type ?Action des autres composants.

Exemple de synchronisation sur CSMA\_CD :

CSMA\_CD est construit à partir de 2 types d'automates :

- L'automate Sender dont les actions sont les suivantes : !begin, !fin, ?CD et ?busy
- L'automate Bus dont les actions sont les suivantes : ?begin, ?fin, !CD et !busy

Le block « Système est construit de 2 automates Sender (indexés 1 et 2) et d'un automate Bus (Il est possible de généraliser ce système à n Sender et un unique Bus).

L'algorithme de synchronisation par rendez-vous crée alors la table suivante :

	Sender (1)	Sender (2)	Bus (3)	Label
<input type="checkbox"/>	!begin	epsilon	?begin	begin
<input type="checkbox"/>	!fin	epsilon	?fin	fin
<input type="checkbox"/>	epsilon	!begin	?begin	begin
<input type="checkbox"/>	epsilon	!fin	?fin	fin
<input type="checkbox"/>	?CD	?CD	!CD	CD
<input type="checkbox"/>	?busy	?busy	!busy	busy

On remarque que la synchronisation entre !begin et ?begin engendre 2 vecteurs (n dans le cas généralisé), alors que la synchronisation entre !CD et ?CD n'engendre qu'un vecteur (comme dans le cas généralisé).

L'algorithme de synchronisation par broadcast crée la table suivante :

	Sender (1)	Sender (2)	Bus (3)	Label
<input type="checkbox"/>	?busy	?busy	!busy	busy
<input type="checkbox"/>	!fin	epsilon	epsilon	fin
<input type="checkbox"/>	epsilon	!begin	epsilon	begin
<input type="checkbox"/>	epsilon	!fin	epsilon	fin
<input type="checkbox"/>	!fin	epsilon	?fin	fin
<input type="checkbox"/>	epsilon	!begin	?begin	begin
<input type="checkbox"/>	?CD	epsilon	!CD	CD
<input type="checkbox"/>	epsilon	epsilon	!busy	busy
<input type="checkbox"/>	?CD	?CD	!CD	CD
<input type="checkbox"/>	?busy	epsilon	!busy	busy
<input type="checkbox"/>	epsilon	?CD	!CD	CD
<input type="checkbox"/>	epsilon	?busy	!busy	busy
<input type="checkbox"/>	!begin	epsilon	?begin	begin
<input type="checkbox"/>	epsilon	epsilon	!CD	CD
<input type="checkbox"/>	!begin	epsilon	epsilon	begin
<input type="checkbox"/>	epsilon	!fin	?fin	fin

La synchronisation entre !CD et ?CD a engendré 4 vecteurs :

- 1 vecteur concerne tous les composants (i.e. ?CD, ?CD, !CD) ce vecteur est identique à celui déduit par l'algorithme par Rendez-Vous
- 1 vecteur ne concerne aucun composant (i.e. epsilon,epsilon, !CD) il correspond au cas ou aucun composant ne reçoit le message

2 vecteurs concernent uniquement 1 des deux composants