ENSTA BRETAGNE
Pôle STIC
2, rue François Verny,
29806 BREST Cedex 9

Philippe Dhaussy,
Jean-Charles Roger
philippe.dhaussy, jean-charles.roger @ensta-bretagne.fr

OBPe (OBP Explorer) V. 1.3

Documentation

htpp://www.obpcdl.org

29 septembre 2012

doc_manual_OBPe_<date>.docx

Sommaire

Résumé

Ce document présente l'utilisation de l'outil OBPe (OBP Explorer) version 1.3.

La version actuelle d'OBPe fonctionne uniquement avec la version du langage CDL 2.0 (voir documentation CDL).

OBPe peut être téléchargé sur le site **http://www.obpcdl.org**.

Notes :

Dans la version actuelle d'OBPe, les patrons de propriétés (*Response, Absence, Existence*) sont parsées et traduits en automates observateurs. Néanmoins, leur exploitation lors de l'exploration par OBP Explorer dans les contextes n'est pas encore opérationnelle (actuellement encore sous développement).

Pour utiliser les observateurs dans les contextes, les propriétés doivent être spécifiées directement sous la forme d'automates (cf la documentation CDL).

1 Installation

Pour installer **OBPe :**

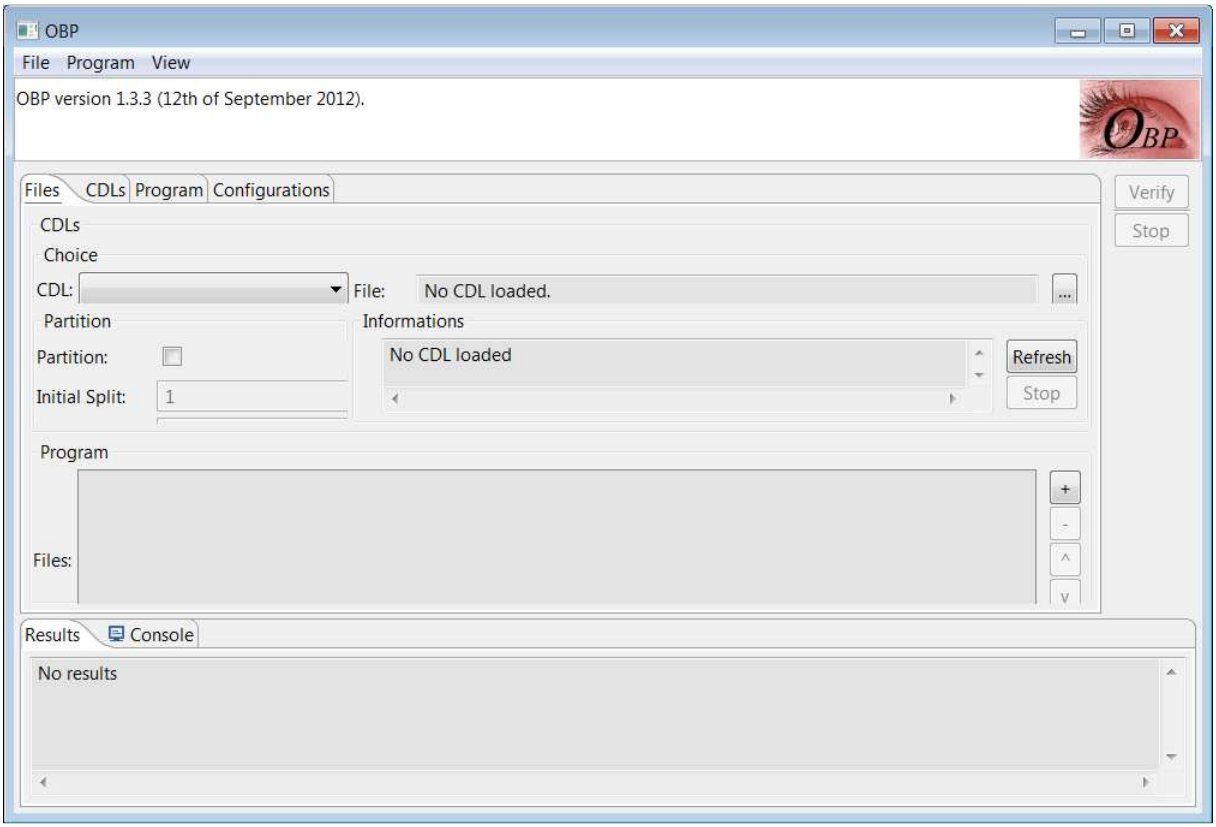
- Télécharger OBPe sur : **http://www.obpcdl.org**
- Décompresser, dans un répertoire de travail, par exemple pour Windows, le fichier **:**

OBP-xxx-win-32.zip

Les fichiers sont copiés dans le répertoire **OBP**.

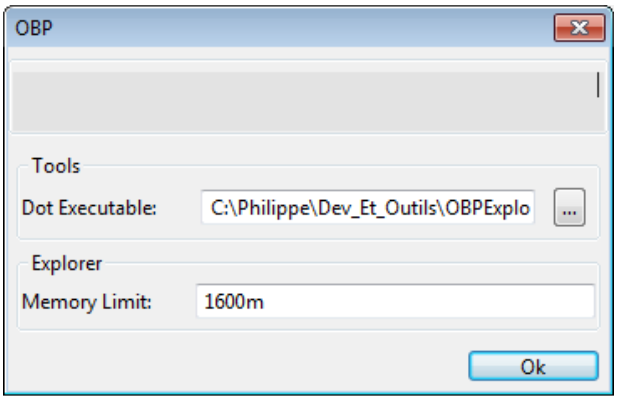
2 Lancement d'OBPe

Dans le répertoire OBP, exécuter : **OBP.exe**
La fenêtre suivante s'ouvre :



3 Paramétrage d'OBPe

Sélectionner le menu **File** puis **Preferences** pour le paramétrage de l'outil.



Champ Tools

- **Dot Executable** : localisation de l'exécutable dot.exe (permet de visualiser les automates Fiacre et les automates observateurs). Par défaut, dot.exe est dans le répertoire OBP/external.

Champ Explorer

- **Memory Limit** : Indique la taille mémoire allouée pour la machine virtuelle. Elle peut être indiquée en mega-octets (m) ou giga-octets (g). Il est possible d'indiquer un pourcentage de la mémoire totale de la machine.

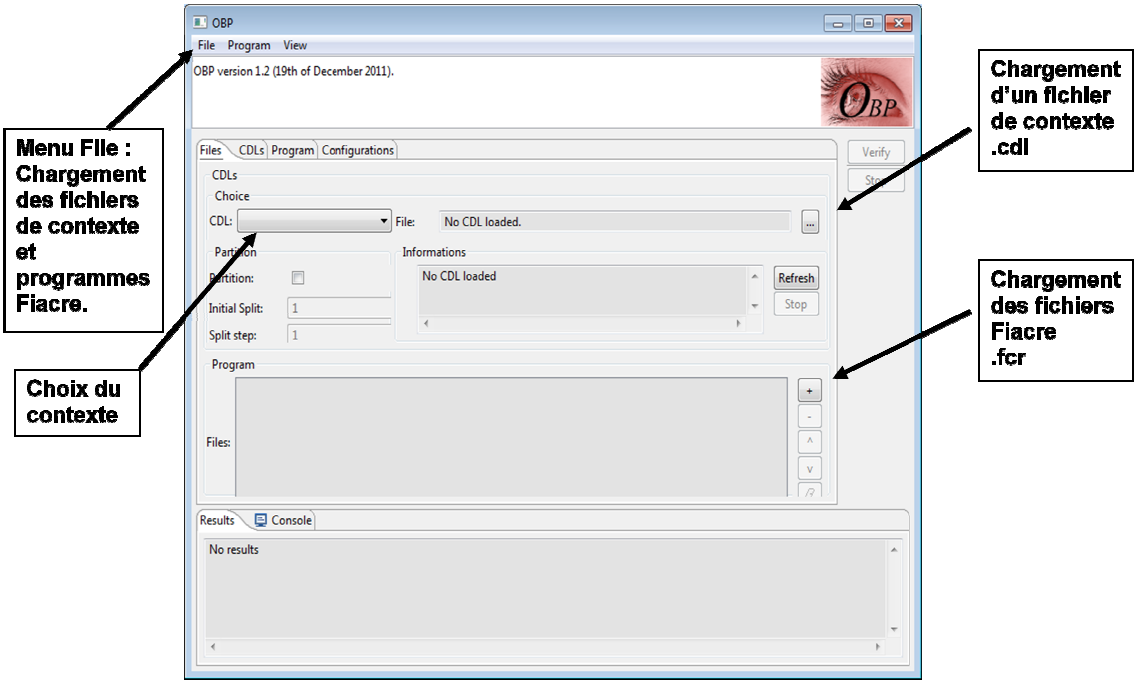
***Attention** : Sous Windows 32bit et Linux 32bit, la taille mémoire allouée maximale est limitée.* Sous Windows, il n'est pas possible d'allouer plus de '1600m'. Sous Linux, il n'est pas possible d'allouer plus de '2600m'.

Les données de paramétrage sont sauvegardées dans le fichier : **.OBP.conf**
Exemple : C:\Utilisateurs\dhaussph**.OBP.conf**

4 Chargement des modèles

Le chargement des fichiers de contexte CDL (.cdl) et Fiacre (.fcr) s'effectue par 2 façons différentes :

- Par le menu **: File**
- Par les boutons à droite de l'interface.



L'onglet **Files** permet de :

- Choisir le contexte pour l'exploration,
- Choisir les paramètres de partitionnement des contextes,

L'onglet **CDLs** permet d'afficher le contenu des contextes CDL et les propriétés sous la forme d'automates observateurs.

L'onglet **Program** permet d'afficher le programme Fiacre

- Soit textuellement. Pour cela, sélectionner dans le menu **View** : **Show program Source**
- Soit graphiquement. Pour cela, désélectionner dans le menu **View** : **Show program Source**

L'onglet **Configuration** permet, suite à une exploration :

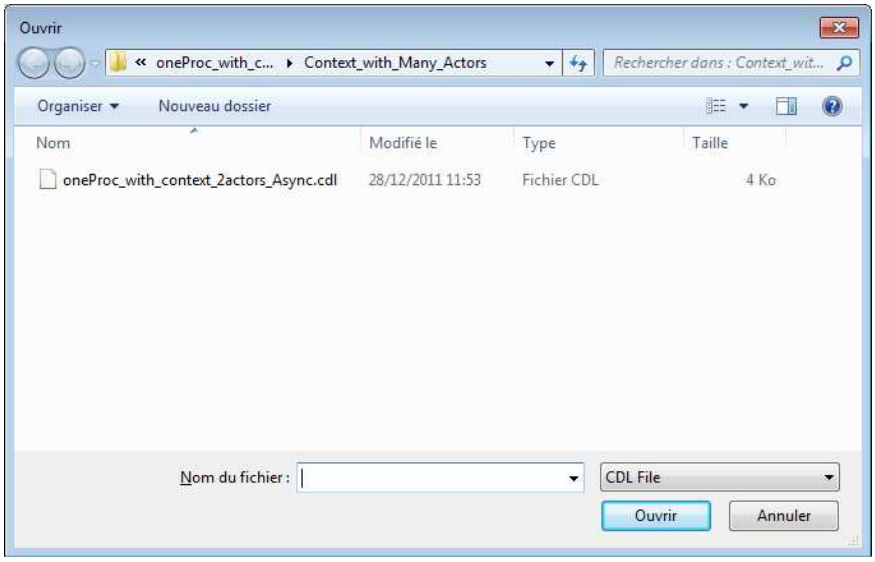
- d'afficher les configurations
- d'afficher un diagramme de séquence
- d'afficher le graphe d'exécution
- de ré-exécuter une exploration.

4.1 Chargement et affichage d'un modèle de contexte CDL

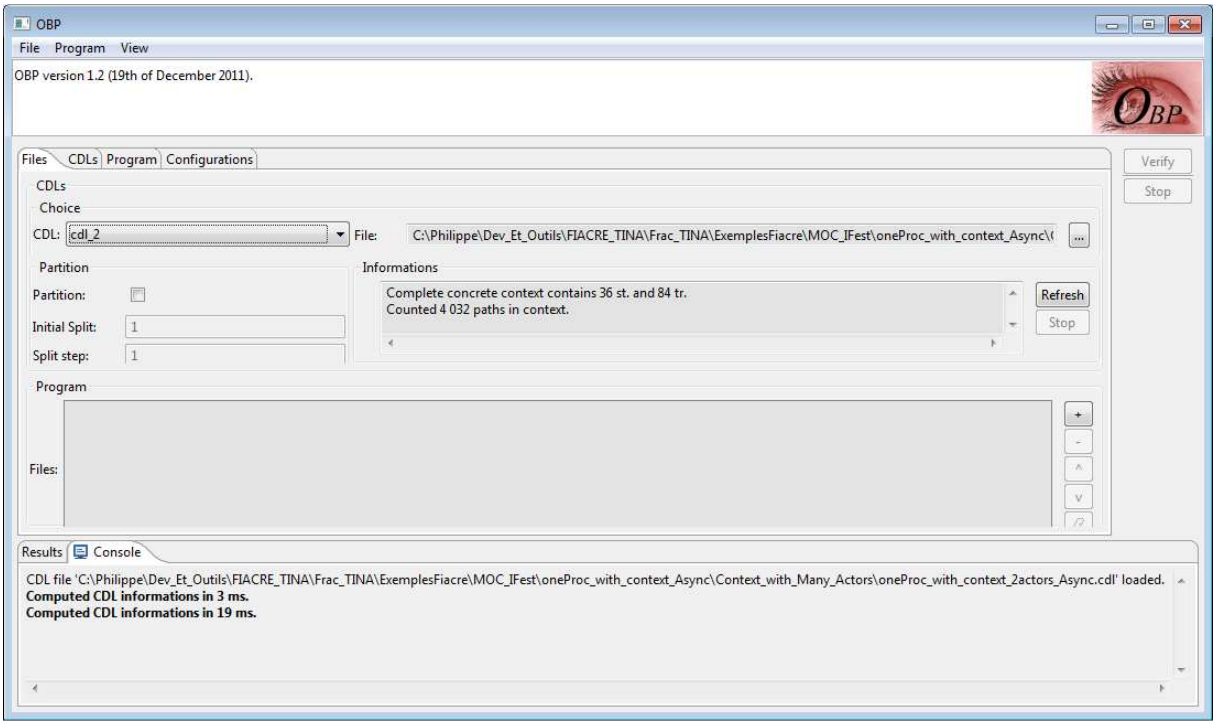
4.1.1 Chargement d'un fichier CDL

Le chargement d'un fichier de contexte (.cdl) peut s'effectuer avec 2 façons :

Menu : File → Open CDL file ou bouton **« … »**



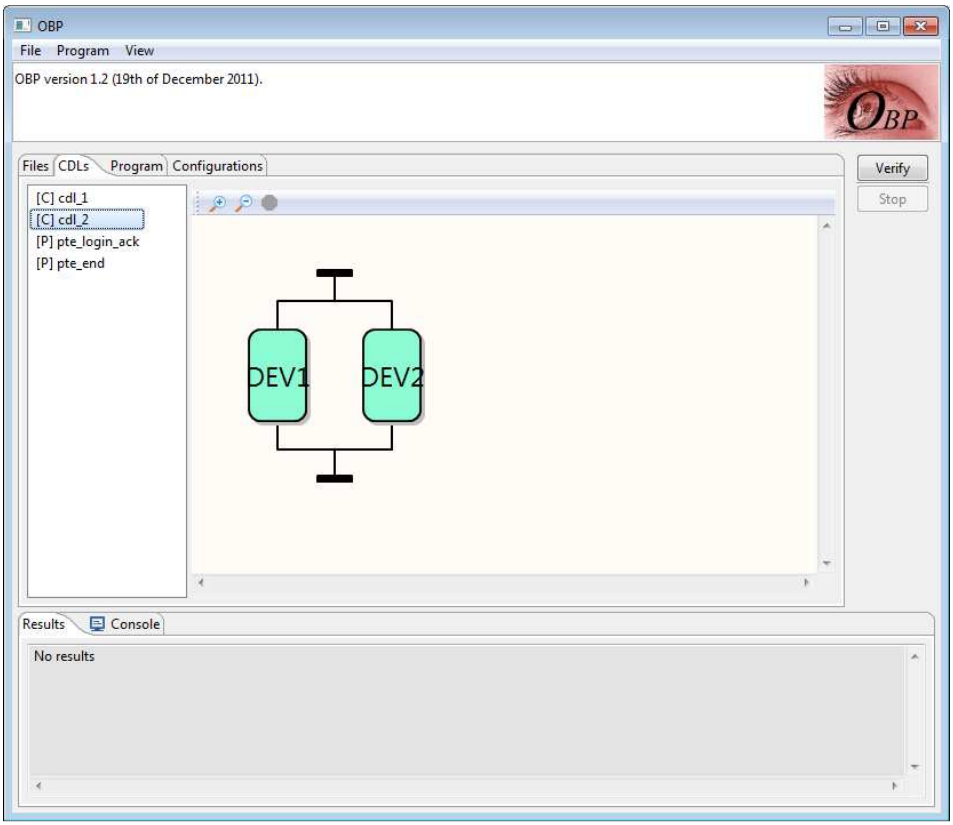
Choisir un ficher CDL. Par exemple : **oneProc_with_context_2actors_Async.cdl**



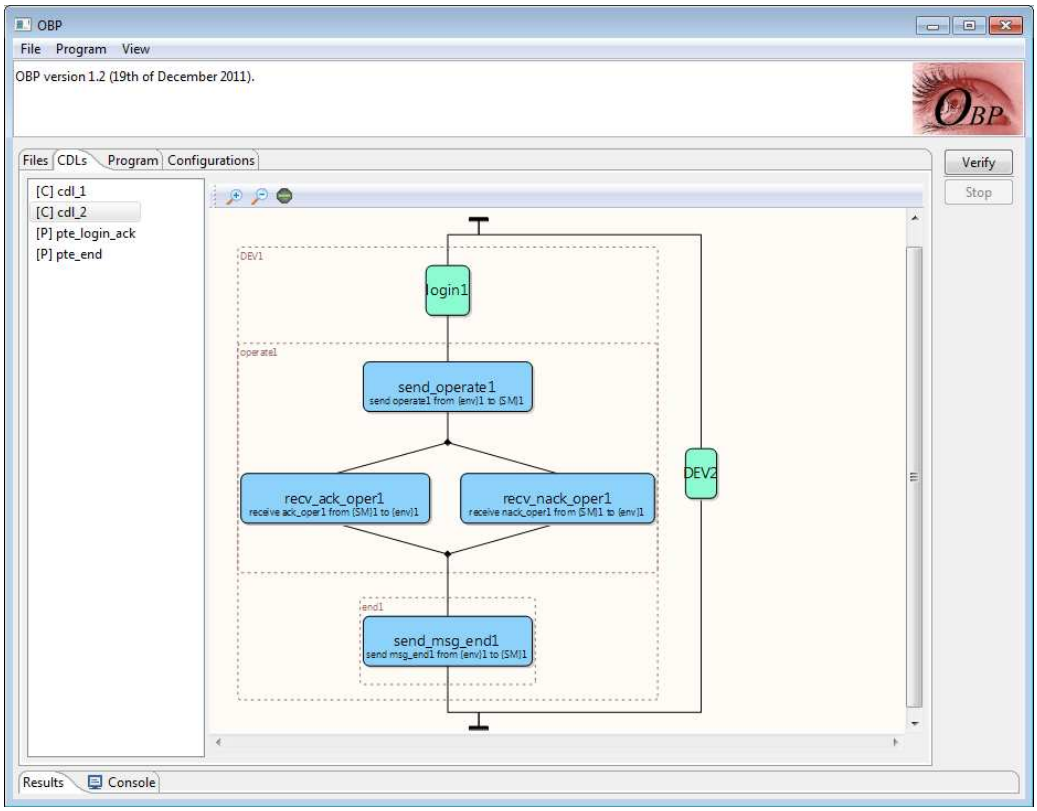
Un fichier CDL peut contenir plusieurs modèles de contexte.
Le champ **Informations** indique le nombre d'états et de transitions associés au contexte chargé (dans l'exemple, le contexte **cdl_2** contient 36 états et 84 transitions).
Le bouton **Refresh** permet de recalculer ces informations, et de calculer le nombre de chemins (linéaires) d'exécution possibles de ce contexte (dans l'exemple, le contexte **cdl_2** contient 4 032 chemins).

4.1.2 Affichage graphique du contexte

Sélectionner l'onglet **CDL.**
Puis sélectionner un des contextes **cdl_1** ou **cdl_2.** Ici, le diagramme du contexte **cdl_2** s'affiche.

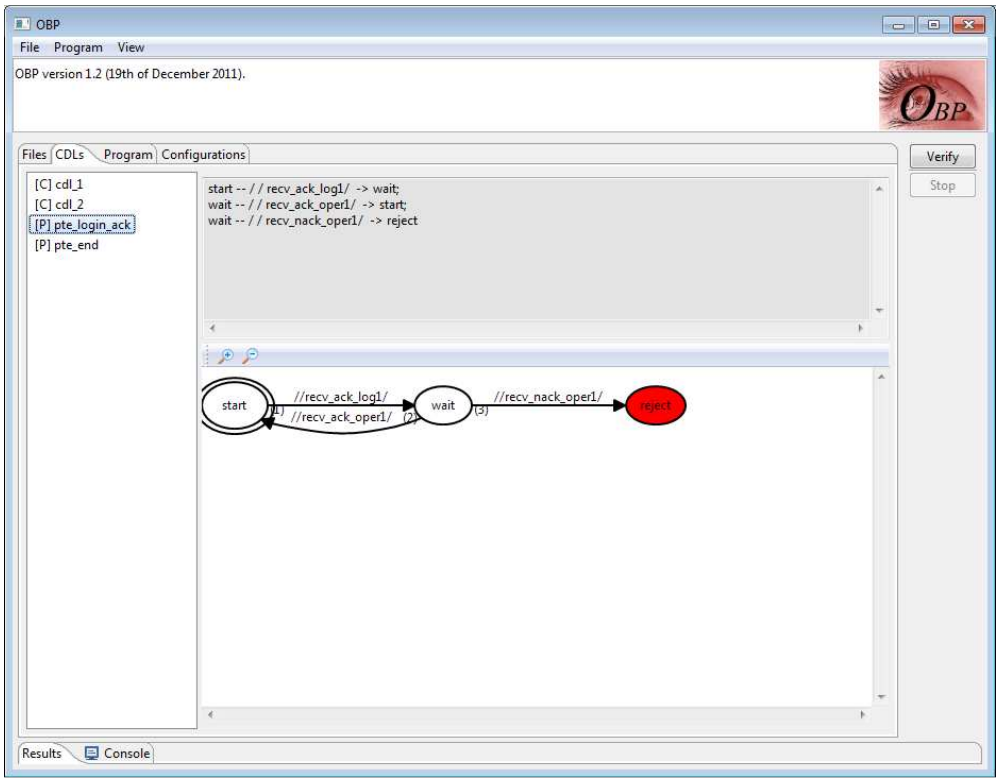


Chaque nœud du modèle CDL peut être affiché et déplié pour indiquer les opérations de communication ***send*** et ***receive***. Par exemple, le nœud **Dev1** est partiellement déplié.



4.1.3 Affichage graphique des observateurs

Sélectionner une des propriétés. Ici ***pte_login_ack*** est sélectionnée.
L'automate observateur est affiché textuellement et graphiquement.

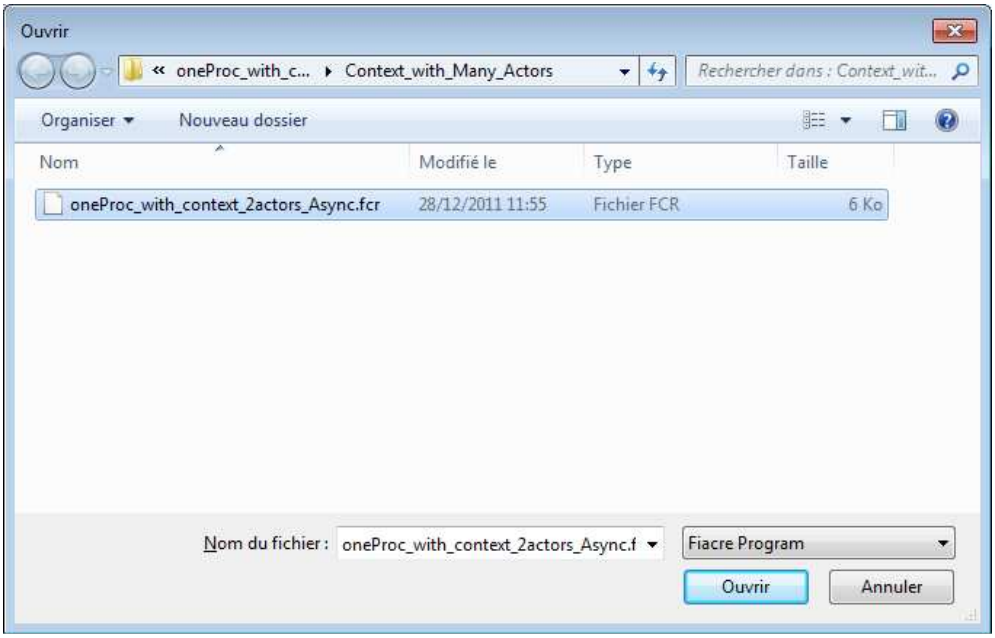


4.2 Chargement et affichage d'un modèle Fiacre

4.2.1 Chargement d'un fichier Fiacre

Le modèle Fiacre est contenu dans un ou des fichiers au format Fiacre (.fcr). Il peut être composé d'un ou de plusieurs fichiers mais au moins un des fichiers doit contenir un composant : **component.**
Le chargement des fichiers Fiacre peut s'effectuer avec 2 façons :

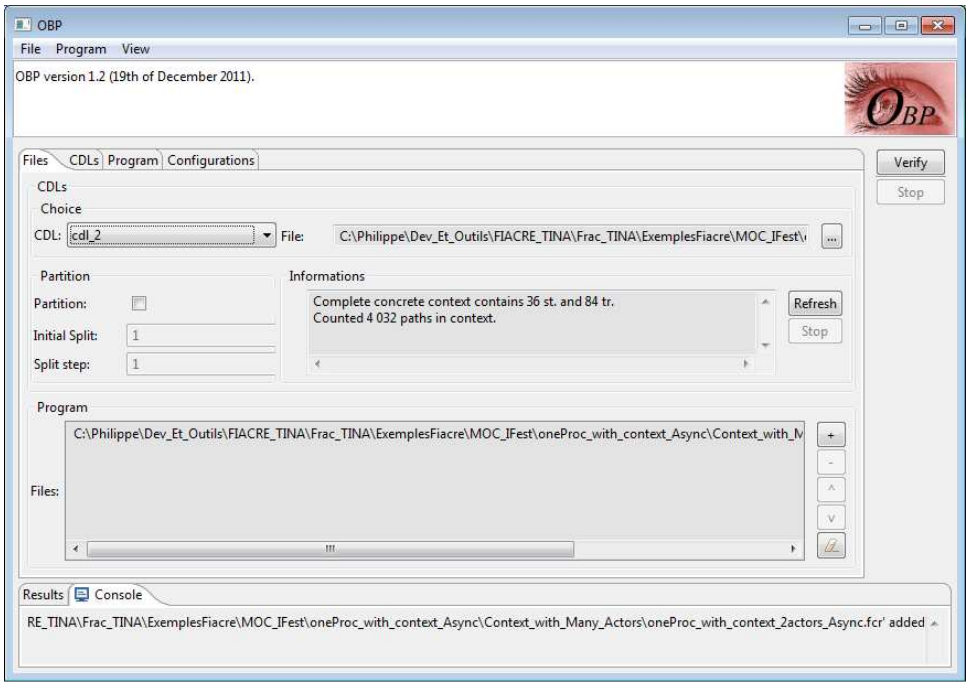
Menu : File → Open program File ou bouton **« + »**



Choisir un ficher fiacre. Par exemple : **oneProc_with_context_2actors_Async.fcr**

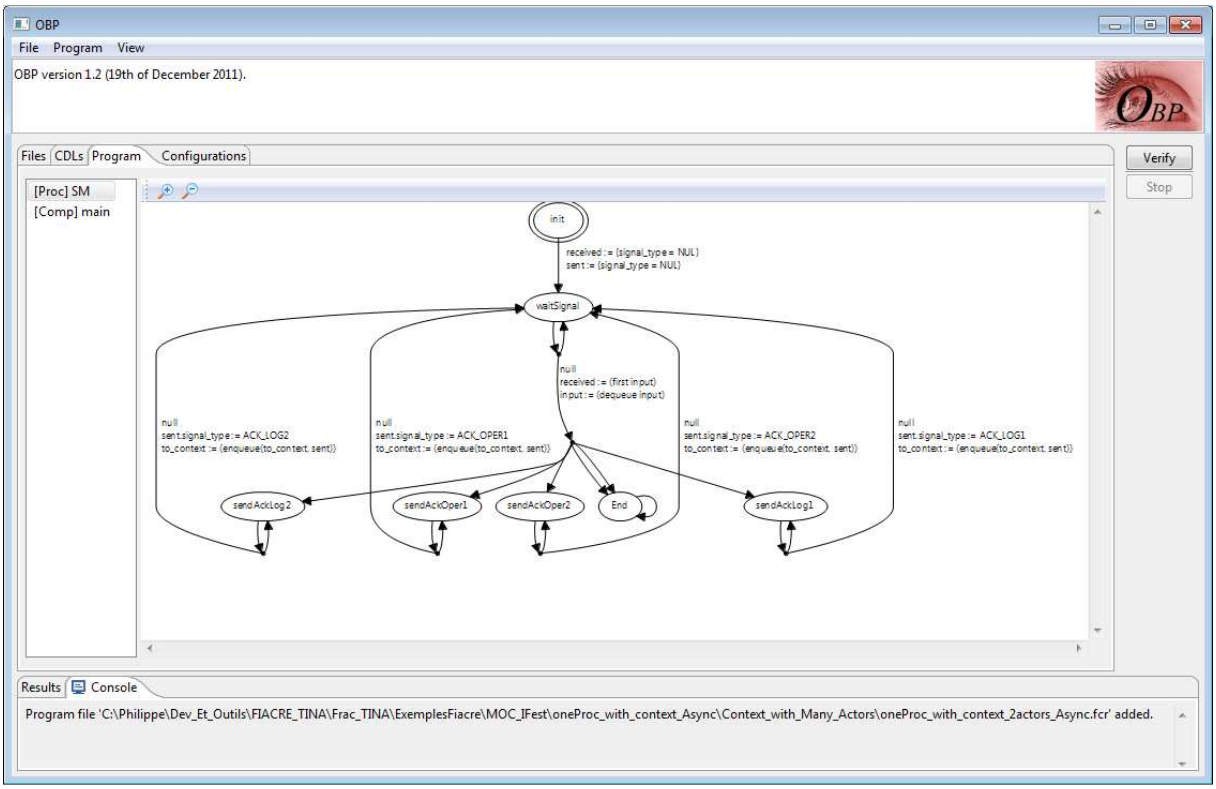
Le programme fiacre est chargé et parsé par le parser de OBPe (similaire à Frac) pour générer des données nécessaires pour les affichages des automates et les analyses de retour de preuves. Si le fichier est modifié après avoir été ouvert dans OBPe, OBPe le recharge automatiquement et affiche dans la **console** : 'Program file reloaded'.

Dans le cas où plusieurs fichiers fiacre composent le modèle Fiacre, OBPe concatène dans le répertoire de travail les fichiers en un seul fichier nommé : **OBPComplete.fcr**.


OBP
File Program View
OBP version 1.2 (19th of December 2011).
Files CDLs Program Configurations
Verify
Stop
CDLs
Choice
CDL: cdl_2
File:
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\
Partition
Partition:
Initial Split: 1
Split step: 1
Informations
Complete concrete context contains 36 st. and 84 tr.
Counted 4 032 paths in context.
Refresh
Stop
Program
Files:
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_M
Results Console
RE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_Actors\oneProc_with_context_2actors_Async.fcr' added

4.2.2 Affichage graphique des automates des processus

Pour afficher les automates des processus Fiacre chargés :
Sélectionner l'onglet **Program** puis sélectionner un des processus (ici : **SM**) :



Les labels visualisés dans chaque transition d'un automate sont les instructions du programme Fiacre pour cette transition.

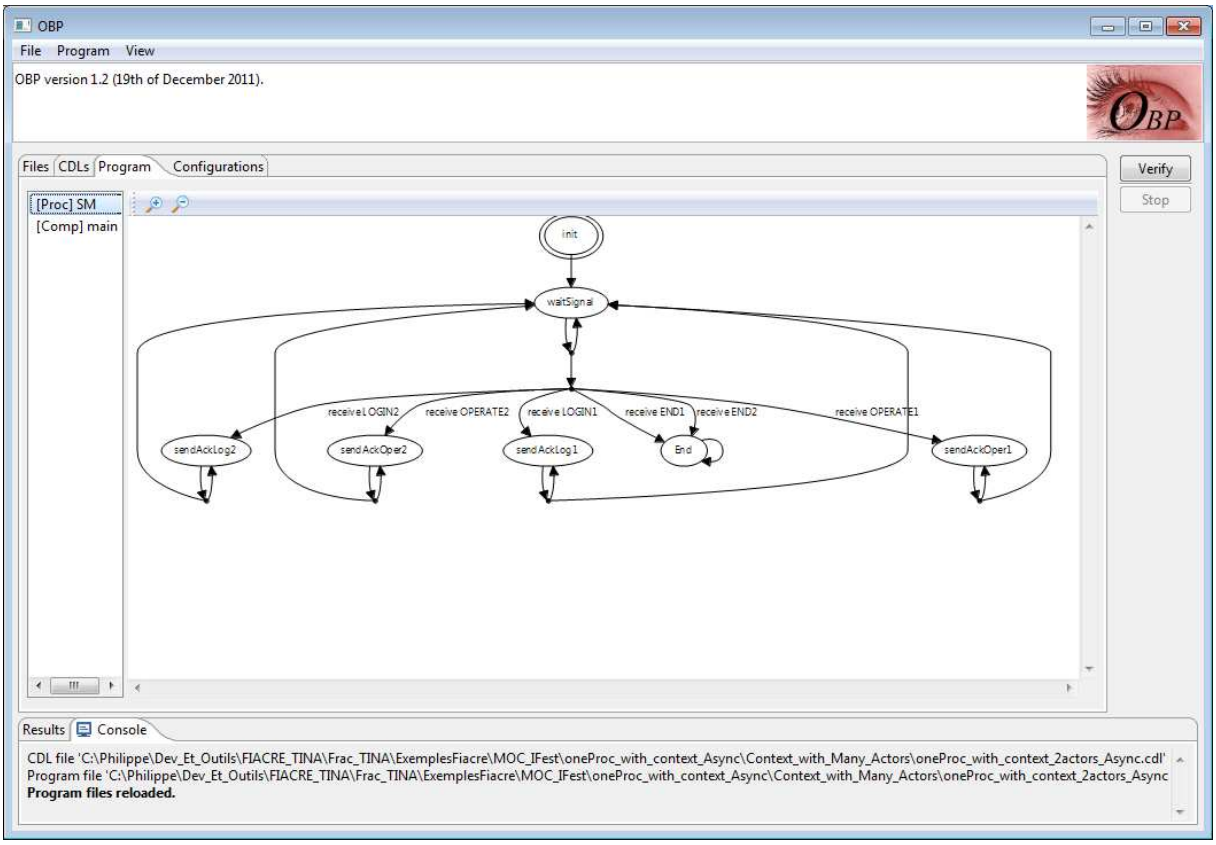
Affichage de commentaires personnalisés dans les transitions des automates affichés

Il est possible d'afficher des commentaires spécifiques dans les transitions d'un automate d'un processus Fiacre, sans afficher les instructions des transitions. Pour cela il faut

1. sélectionner dans le menu : **View : Use description comments in graphs**
2. Et ajouter le commentaire dans le code de la transition avec le format suivant :

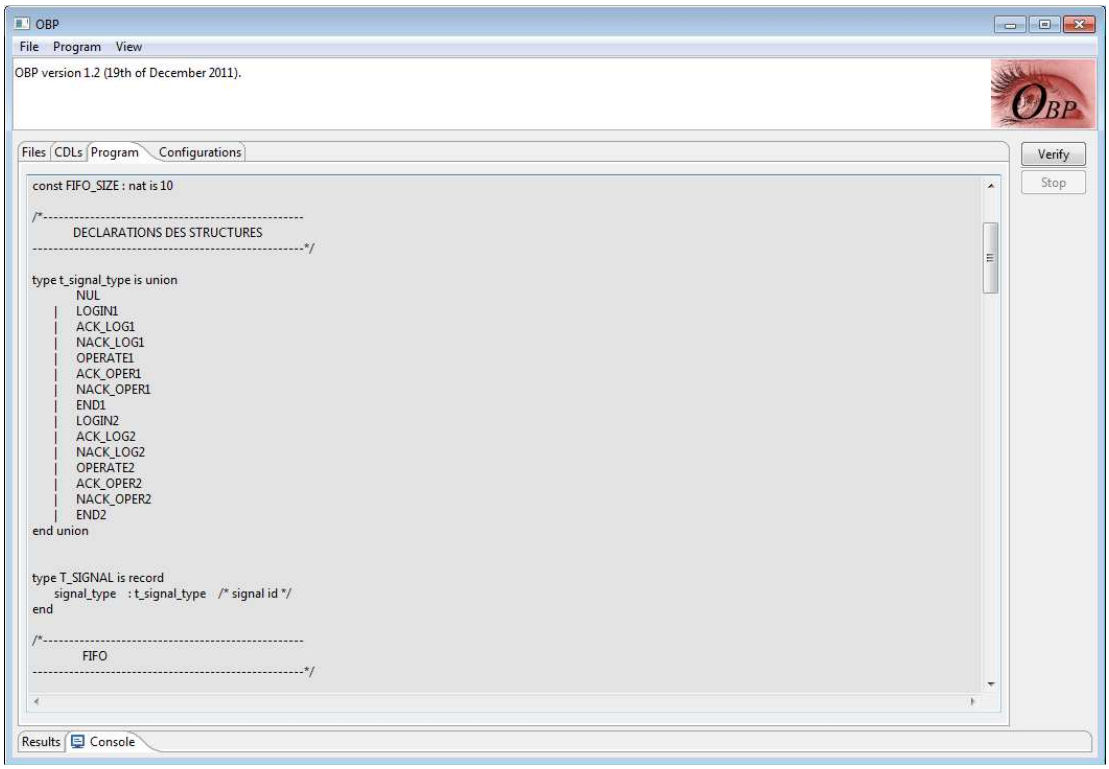
/*@ commentaire a afficher */

Exemple : /*@ receive LOGIN1 */



4.2.3 Affichage du code des automates des processus

Dans le menu : **View,** sélectionner : **Show Program Source.**
Le code du programme Fiacre s'affiche.



Pour revenir à l'affichage des automates :
Dans le menu : **View,** désélectionner : **Show Program Source.**

4.2.4 Composition d'un programme Fiacre et d'un CDL

Dans le programme Fiacre, le composant principal doit être un ***component*** et doit contenir les variables partagées entre les processus Fiacre et le contexte décrit dans le fichier CDL.

Le composant principal doit posséder une déclaration d'une file de messages (*queue*) nommée ***toContext***. Si le contexte CDL est vide, la déclaration de la file ***toContext*** n'est pas nécessaire***.*** Cela peut être le cas où le fichier CDL ne contient que la déclaration des évènements, prédicats et propriétés observateurs mais pas de contexte.

Le composant doit aussi posséder une déclaration de file pour chaque instance de processus référencé dans le CDL. Par exemple, un message envoyé depuis un CDL à l'instance de processus **{SM}1** sera ajouté à la file **SM_1** du composant principal.

5 Exécution de l'exploration et la vérification

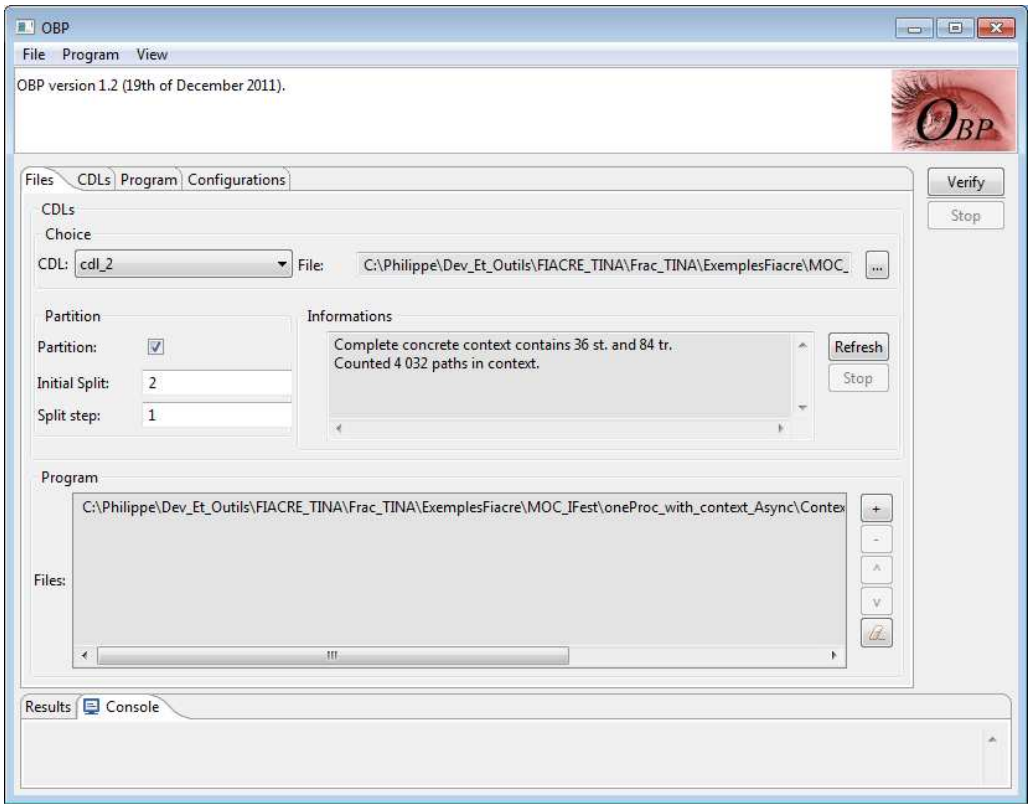
5.1 Principes du partitionnement du contexte

- Un contexte entier peut être composé avec le modèle Fiacre. Dans ce cas, OBPe compose le modèle Fiacre avec le contexte.

- En cas d'explosion de l'espace d'états lors de l'exploration par OBPe, ce contexte peut être partitionné en un ensemble de graphes de contexte. OBPe lance alors une exploration pour chaque graphe de contexte composé avec le modèle Fiacre.

Pour partitionner le contexte, il faut

- cocher la case **Partition**
- préciser le coefficient **Initial Split**
- préciser le coefficient **Split step**



Ces coefficients correspondent aux profondeurs, dans le graphe du contexte à partitionner, prises en compte lors du partitionnement. Plus ces coefficients sont élevés, plus le nombre de partitions de contexte générés par OBPe sera élevé. **Initial Split** est le coefficient pris en compte lors la 1ère étape de partitionnement. **Split step** est le coefficient pris en compte lors des étapes suivantes de partitionnement.
En cas de partitionnement (case **Partition** cochée), OBP partitionne automatiquement et récursivement les contextes qui amènent à une explosion du nombre de configurations lors de l'exploration avec le modèle du système.

5.2 Lancement de la compilation et de l'exploration du programme Fiacre

Menu **: Program → Verify**
ou bouton **Verify (**à droite**)**

Le programme fiacre est chargé et parsé par Frac pour générer les données pour OBP Explorer qui explore ensuite le programme Fiacre composé avec son contexte. La trace de l'exploration s'affiche dans la **Console**.
Pour arrêter l'exploration en cours : bouton **Stop** à droite de l'interface.

5.2.1 Exploration sans partitionnement

L'exemple suivant montre la trace de prise en compte d'un contexte : compilation et exploration avec le contexte **cdl_2** sans partitionnement.

```
[Explore, started at 28 déc. 2011 13:11:06]
Compiling Fiacre file
'C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many
_Actors\oneProc_with_context_2actors_Async.fcr'.
```

```
Exec: [ecj  -1.6 -d
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\bin
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\java]
Exec: [C:\Program Files (x86)\Java\jre6\bin\java.exe -Xms1600m -Xmx1600m -XX:+UseParallelGC -XX:NewRatio=20 -classpath
;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP\jars;C:\Philippe\Dev_Et_Outils\OBPExpl
orer\OBP\jars\OBP.jar obp.explorer.Explorer --boost --cc - --program-path
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\bin --out
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\confs\cdl_2.confs
oneproc_with_context_2actors_async.OneProc_with_context_2actors_AsyncRoot]
Process last 0.428s and exit with 0.
Explored 161 states and 238 transitions in 0,428 seconds.
[Explore, ended at 28 déc. 2011 13:11:07]
```

Une synthèse de l'exploration est indiquée dans l'onglet **Results** :

Exemple :

```
--------------------Informations------------------
Exploration: 28 déc. 2011 13:37:29
CDL 'cdl_2' in file:
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.cdl.
Verified properties : 'pte_login_ack' 'pte_end'.
Fiacre file names:
-
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.fcr.

--------------------Synthesis---------------------
Explored 161 states and 238 transitions in 0,309 seconds.
Context 'cdl_2' contained 36 states, and 84 transitions.
Result is stored in file
'C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many
_Actors\oneProc_with_context_2actors_Async.obp\confs\cdl_2.confs'.

--------------------Verification---------------------
Properties:
- '{pte_end}1':  reached success the first time in configuration 67.

--------------------Warning----------------------
No warning.
```

La trace de résultat indique la liste des propriétés qui sont pris en compte dans l'exploration. Ici, les propriétés : ***pte_login_ack*** et ***pte_end***.

Le résultat de la vérification des propriétés est indiqué :
La propriété ***pte_end*** est en succès pour la configuration **67**.

Le champ « Warning » indique si un des observateurs n'a pas été sensibilisé, c'est-à-dire n'a pas quitté son état ***start***).

5.2.2 Exploration avec partitionnement

L'exemple suivant montre la trace de prise en compte d'un contexte (compilation et exploration avec le contexte **cdl_2** avec partitionnement (Initial Split : 2, Split step : 1). 6 sous-contextes sont générés pour les explorations.

```
[Explore, started at 28 déc. 2011 13:46:35]
-----------------------------------------------------
---> Exploring for context 'cdl_2_0' (26 states and 54 transitions).
Exec: [C:\Program Files (x86)\Java\jre6\bin\java.exe -Xms1600m -Xmx1600m -XX:+UseParallelGC -XX:NewRatio=20 -classpath
;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP\jars;C:\Philippe\Dev_Et_Outils\OBPExpl
orer\OBP\jars\OBP.jar obp.explorer.Explorer --boost --cc - --program-path
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\bin --out
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\confs\cdl_2_0.confs
oneproc_with_context_2actors_async.OneProc_with_context_2actors_AsyncRoot]
Process last 0.256s and exit with 0.
Explored 101 states and 143 transitions in 0,256 seconds.
---> Exploration of context 'cdl_2_0' succeeded.
-----------------------------------------------------
---> Exploring for context 'cdl_2_1' (26 states and 54 transitions).
Exec: [C:\Program Files (x86)\Java\jre6\bin\java.exe -Xms1600m -Xmx1600m -XX:+UseParallelGC -XX:NewRatio=20 -classpath
;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP\jars;C:\Philippe\Dev_Et_Outils\OBPExpl
orer\OBP\jars\OBP.jar obp.explorer.Explorer --boost --cc - --program-path
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\bin --out
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\confs\cdl_2_1.confs
oneproc_with_context_2actors_async.OneProc_with_context_2actors_AsyncRoot]
Process last 0.218s and exit with 0.
Explored 5 states and 4 transitions in 0,218 seconds.
---> Exploration of context 'cdl_2_1' succeeded.
-----------------------------------------------------
---> Exploring for context 'cdl_2_2' (27 states and 62 transitions).
Exec: [C:\Program Files (x86)\Java\jre6\bin\java.exe -Xms1600m -Xmx1600m -XX:+UseParallelGC -XX:NewRatio=20 -classpath
;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP\jars;C:\Philippe\Dev_Et_Outils\OBPExpl
orer\OBP\jars\OBP.jar obp.explorer.Explorer --boost --cc - --program-path
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\bin --out
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\confs\cdl_2_2.confs
oneproc_with_context_2actors_async.OneProc_with_context_2actors_AsyncRoot]
Process last 0.222s and exit with 0.
Explored 87 states and 119 transitions in 0,222 seconds.
---> Exploration of context 'cdl_2_2' succeeded.
-----------------------------------------------------
---> Exploring for context 'cdl_2_3' (27 states and 62 transitions).
Exec: [C:\Program Files (x86)\Java\jre6\bin\java.exe -Xms1600m -Xmx1600m -XX:+UseParallelGC -XX:NewRatio=20 -classpath
;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP\jars;C:\Philippe\Dev_Et_Outils\OBPExpl
orer\OBP\jars\OBP.jar obp.explorer.Explorer --boost --cc - --program-path
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\bin --out
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\confs\cdl_2_3.confs
oneproc_with_context_2actors_async.OneProc_with_context_2actors_AsyncRoot]
Process last 0.22s and exit with 0.
Explored 87 states and 119 transitions in 0,22 seconds.
---> Exploration of context 'cdl_2_3' succeeded.
-----------------------------------------------------
---> Exploring for context 'cdl_2_4' (26 states and 54 transitions).
Exec: [C:\Program Files (x86)\Java\jre6\bin\java.exe -Xms1600m -Xmx1600m -XX:+UseParallelGC -XX:NewRatio=20 -classpath
;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP\jars;C:\Philippe\Dev_Et_Outils\OBPExpl
orer\OBP\jars\OBP.jar obp.explorer.Explorer --boost --cc - --program-path
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\bin --out
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\confs\cdl_2_4.confs
oneproc_with_context_2actors_async.OneProc_with_context_2actors_AsyncRoot]
Process last 0.237s and exit with 0.
```

```
Explored 101 states and 143 transitions in 0,237 seconds.
---> Exploration of context 'cdl_2_4' succeeded.
-----------------------------------------------------
---> Exploring for context 'cdl_2_5' (26 states and 54 transitions).
Exec: [C:\Program Files (x86)\Java\jre6\bin\java.exe -Xms1600m -Xmx1600m -XX:+UseParallelGC -XX:NewRatio=20 -classpath
;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP;C:\Philippe\Dev_Et_Outils\OBPExplorer\OBP\jars;C:\Philippe\Dev_Et_Outils\OBPExpl
orer\OBP\jars\OBP.jar obp.explorer.Explorer --boost --cc - --program-path
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\bin --out
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_Many_
Actors\oneProc_with_context_2actors_Async.obp\confs\cdl_2_5.confs
oneproc_with_context_2actors_async.OneProc_with_context_2actors_AsyncRoot]
Process last 0.22s and exit with 0.
Explored 5 states and 4 transitions in 0,22 seconds.
[Explore, ended at 28 déc. 2011 13:46:36]
```

Une synthèse de l'exploration est indiquée dans l'onglet **Results** :

Exemple :

```
--------------------Informations--------------------
Exploration: 28 déc. 2011 13:46:37
CDL 'cdl_2' in file:
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_
Many_Actors\oneProc_with_context_2actors_Async.cdl.
Verified properties : 'pte_login_ack' 'pte_end'.
CDL has been split (initial: 2, step: 1).

Fiacre file names:
-
C:\Philippe\Dev_Et_Outils\FIACRE_TINA\Frac_TINA\ExemplesFiacre\MOC_IFest\oneProc_with_context_Async\Context_with_
Many_Actors\oneProc_with_context_2actors_Async.fcr.

--------------------Synthesis----------------------
-> With split contexts:
Explored 386 states and 532 transitions in 1,373 seconds.

-> Only efficient contexts:
Explored 386 states and 532 transitions in 1,373 seconds.

Total number of contexts: 6
Number of efficient contexts: 6
Number of split contexts: 0.

--------------------Verification---------------------
Properties:
-'{pte_end}1':  reached success the first time in context 'cdl_2_0'.

--------------------Contexts---------------------
- cdl_2_0(26 states and 54 transitions) Explored in 0,256 seconds: 101 states and 143 transitions.
- cdl_2_1(26 states and 54 transitions) Explored in 0,218 seconds: 5 states and 4 transitions.
- cdl_2_2(27 states and 62 transitions) Explored in 0,222 seconds: 87 states and 119 transitions.
- cdl_2_3(27 states and 62 transitions) Explored in 0,22 seconds: 87 states and 119 transitions.
- cdl_2_4(26 states and 54 transitions) Explored in 0,237 seconds: 101 states and 143 transitions.
- cdl_2_5(26 states and 54 transitions) Explored in 0,22 seconds: 5 states and 4 transitions.
```

La trace de résultat indique la liste des propriétés qui sont pris en compte dans l'exploration. Ici, les propriétés : ***pte_login_ack*** et ***pte_end***.

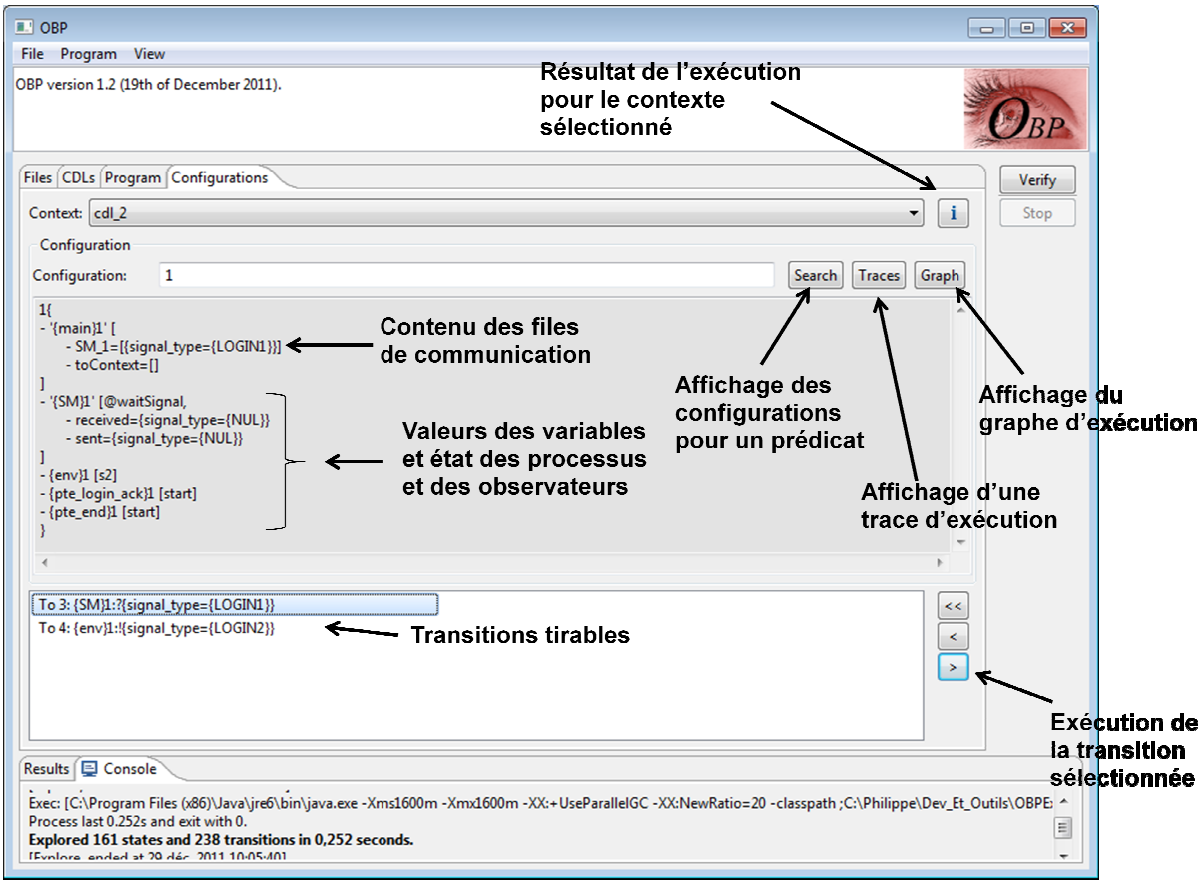
Le résultat de la vérification des propriétés est indiqué : La propriété ***pte_end*** est en succès pour le premier sous-contexte cdl_2_0.

L'onglet **Configurations** permet de connaître le résultat de l'exploration pour les différents sous-contextes issus du partitionnement (voir paragraphe *Analyse du graphe d'exploration*).

5.3 Analyse du graphe d'exploration

Suite à l'exploration, il est possible de consulter l'état des configurations et de ré-exécuter la simulation pas à pas.
Pour cela, indiquer dans le champ **Configuration** le numéro de la configuration que l'on souhaite visualiser.

Par exemple : affichage de la configuration 1 :

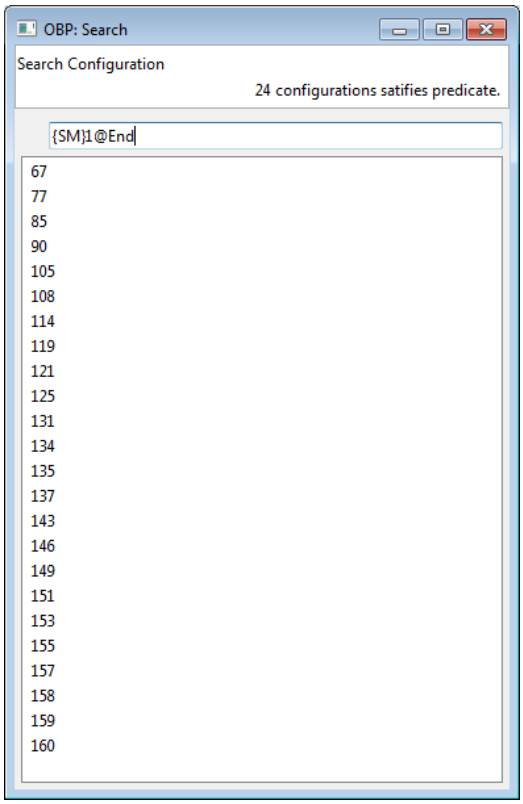


Bouton '>' : Sélectionne la configuration cible de la transition sélectionnée.

Bouton '<' : Sélectionne la configuration la plus proche de la configuration sélectionnée qui possède une transition qui permet de l'atteindre. Cliquer plusieurs fois sur ce bouton permet de revenir à la configuration initiale (0). La configuration est recherchée à partir du numéro de la configuration sélectionnée décrémentée de 1, jusqu'à trouver la première configuration qui contient une transition vers la configuration sélectionnée.

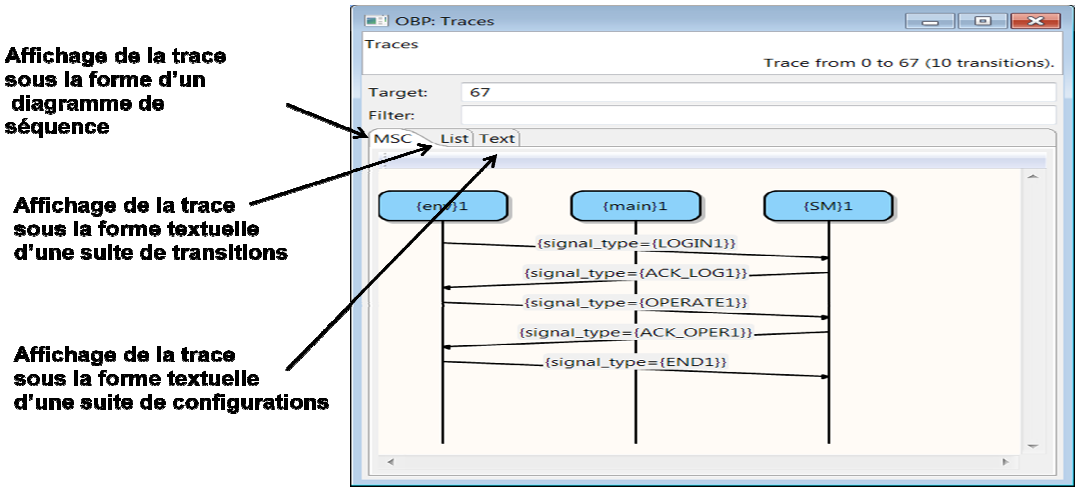
Bouton '<<' : La configuration est recherchée à partir de l'id 0 (calcul plus long).

Bouton Search : Permet de chercher des configurations en fonction de la valeur d'un prédicat. Celui-ci porte sur des valeurs de variables et des états des processus (cf la documentation CDL). Cliquer sur '**Search'** force le chargement des toutes les configurations en mémoire (peut être long).



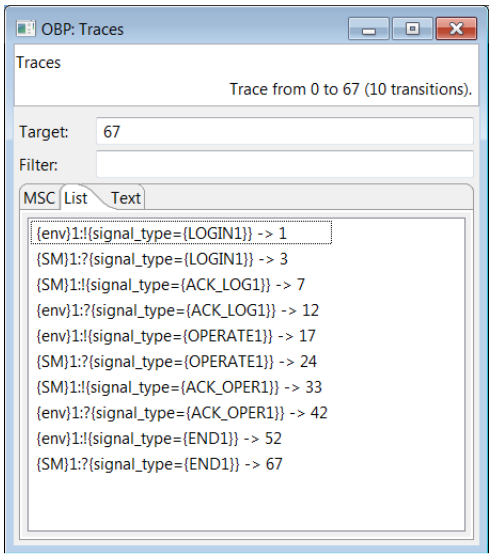
Ici, s'affiche la liste des configurations où le prédicat **{SM}1@End** est vrai.

Bouton Traces : Affiche une des traces (sous la forme d'un diagramme de séquences) à partir de la configuration initiale 0 et menant à la configuration indiquée dans le champ *Target*. L'affichage d'une trace force le chargement des toutes les configurations en mémoire (peut être donc long).

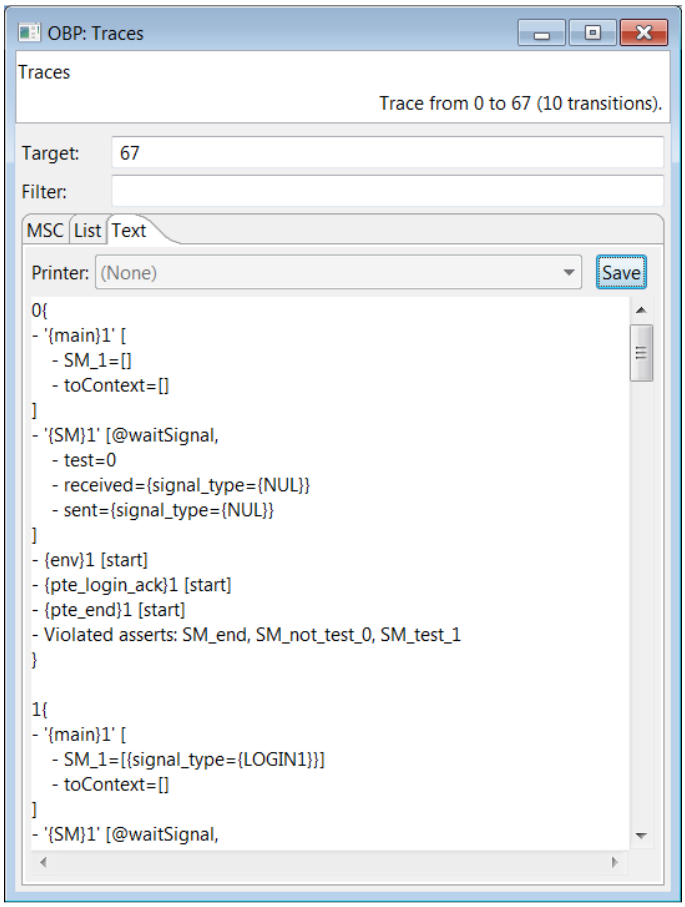


La trace visualisée est la trace la plus rapide à calculer (complexité linéaire). Elle est la même que celle affichée par le clic successif sur le bouton '**<**'.
Un double clic sur l'icône du processus permet de faire disparaître et ré-apparaître sa ligne de vie dans le diagramme.

La trace peut être affichée sous forme textuelle par une liste des transitions : bouton **List.**



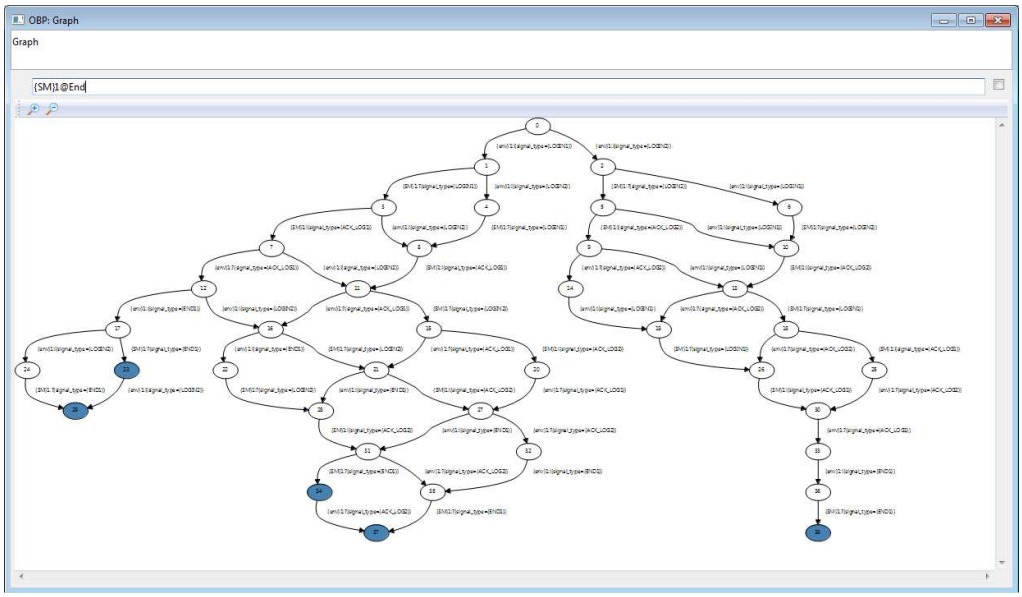
La trace peut être affichée sous forme textuelle par une liste des configurations : bouton **Text.**



La liste des configurations indiquent, pour chaque configuration atteinte lors de l'exécution (trace), l'état des processus, des automates observateurs, des fifos partagées, la valeur de toutes les variables.

Le bouton **Save** permet de sauvegarder la liste des configurations dans un fichier pour un traitement externe éventuel à OBP.

Bouton Graph : Affiche le graphe des configurations (pour l'affichage de graphes de taille limitée).



Le champ d'édition permet d'indiquer un prédicat (ici **{SM}1@End**). Si ce prédicat est vrai, les nœuds des configurations concernées sont colorisés.