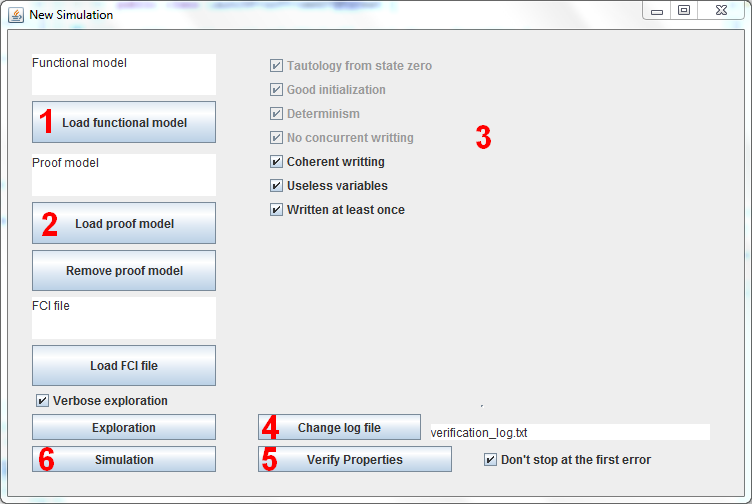
# Manuel d’utilisation de l’outil de preuve 2014 en Java

## Simulation

L’outil de simulation utilisant les fichiers en format 6 lignes se lance à l’aide de « LaunchSimulatorGUI ». La fenêtre suivante s’ouvre alors.d

**1** : Le bouton permet de charger le modèle fonctionnel depuis un format 6 lignes.

**2** : le bouton permet de charger le modèle de preuve depuis un format 6 lignes.

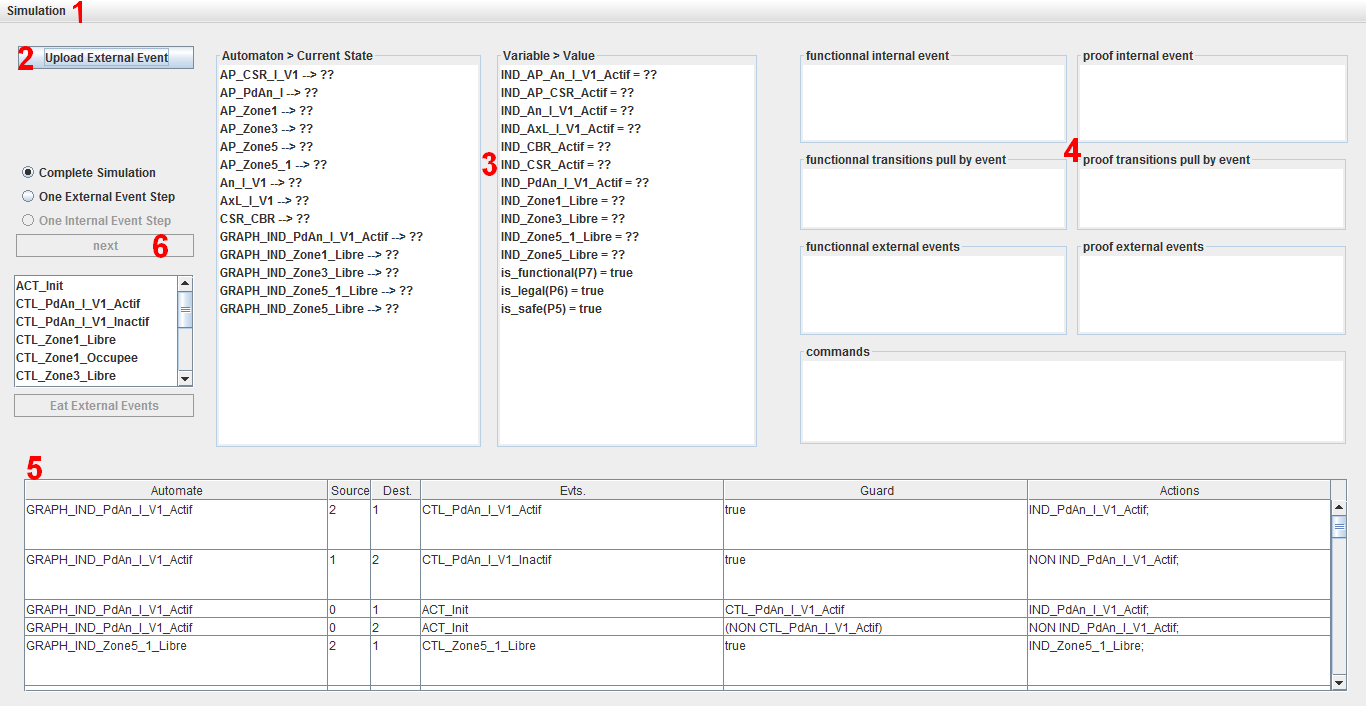
**3** : Ce cadre permet de choisir les vérifications à réaliser sur les modèles. Les descriptions détaillées sont données dans les commentaires ou la documentation de l’outil. Les vérifications grisés ne peuvent pas être désactivés. Il est conseillé de ne rien désactiver.

**4** : Le logiciel écrit tous ses résultats dans la console d’Eclipse dans le cas où le logiciel est lancé avec Eclipse. Les résultats sont également écrits dans un fichier, qui par défaut se nomme « verification\_log.txt ». Ce bouton permet de modifier le fichier de log.

**5** : Lance la vérification des propriétés sélectionnées en 3. Il faudra regarder dans la console ou dans le fichier de log pour avoir le détail des résultats des vérifications. La case « Don’t stop at the first error » permet de trouver le maximum d’erreurs. Désactiver cette case rend la vérification plus rapide car celle-ci s’arrêtera à la première erreur détectée lors de chaque vérification individuelle. Le temps de vérification étant négligeable, il est conseillé de laisser cette case cochée.

**6** : Permet de lancer le simulateur. Il est conseillé de lancer la simulation uniquement lorsque les vérifications passent. En particulier, si la vérification sur l’initialisation ne passe pas, le simulateur pourra crasher lorsqu’on lui demandera de faire des opérations qu’il ne peut pas réaliser.

Une fois la simulation lancée, une nouvelle fenêtre s’ouvre. Le simulateur n’a pas la même sémantique que le code C, en particulier au niveau des FCI. Voir le rapport détaillé pour plus d’information. La seule chose à retenir est qu’aucune garantie n’existe quant à la correspondance entre l’exécution en C et l’exécution du prouveur.

**1** : Le menu permet d’initialiser le modèle, de relancer la simulation ou de lancer une nouvelle simulation (ferme la simulation courante dans ce cas). Pour initialiser le modèle, il suffit de sélectionner la valeur de vérité des CTL. L’initialisation se déroule ainsi : on associe à chaque CTL la valeur de vérité sélectionnée, on injecte l’évènement ACT\_Init, puis on enlève les CTL du modèle. Tant que l’initialisation n’est pas effectuée, il n’est pas possible d’injecter des évènements dans le modèle.

**2** : Ce bouton permet de charger une liste d’évènement externe pour jouer un scénario. Le fichier doit contenir un seul évènement externe par ligne.

**3** : Les deux blocs à coté du numéro 3 affichent respectivement les états des automates et les états des variables. Tant qu’une donnée n’est pas définie des « ?? » remplacent la valeur.

**4** : Ces blocs donnent des informations sur les files, les transitions qui ont été tirées lors de l’exécution ainsi que les commandes générées. Les titres en haut de chaque zone de texte permettent de détailler la fonction de chaque zone.

**5** : Affiche les transitions contenues dans le modèle fonctionnel et le modèle de preuve.

6 : Pour exécuter un évènement, il faut d’abord l’injecter (bouton « Eat »), puis exécuter le modèle. A l’heure où cette documentation est écrite, on peut soit exécuter tous les évènements externes injectés (« Complete Simulation »), soit exécuter le premier évènement de la file des évènements externes (« One External Event Step »).