

Scénarios de tests d'intégration/fonctionnels



Partenaire : Espresso

Nom de l'outil : Editeur Polychrony

Date test : 24/04/2009

Version de l'outil : OpenEmbeDD 1.0.0 Titan

Plateforme de test (OS + processeur) : Fedora Release 8 (noyau 2.6.25.14-69.fc8) Intel

Organisme testeur : OpenEmbeDD integration team

Nom du testeur : Christian Brunette

Récapitulatif des tests

Commentaires du testeur.....	1
Scénario A : perspective Topcased.....	2
Scénario B : création d'un modèle.....	3
Scénario C: installation des exemples.....	4
Scénario D: appel des fonctions du compilateur.....	5
Scénario E : aide en ligne.....	7

Commentaires du testeur

Scénario A : perspective Topcased

Remarques : R.A.S.

Action	Résultat attendu	OK	Commentaires
Aller dans <i>Windows >> Open Perspective >> Other</i>	Une perspective Topcased doit apparaître	Oui	
Choisir la perspective Topcased	Elle doit s'activer (apparaît en haut à droite)	Oui	

Scénario B : création d'un modèle

Action	Résultat attendu	OK	Commentaires
Choisir la perspective Topcased	Elle doit apparaître en haut à droite.	Oui	
Créer un projet Topcased	Projet créé	Oui	
Dans ce projet, clic droit puis <i>New >> Sme diagrams</i>	Modèle par défaut ouvert dans l'éditeur graphique	Oui	
Ajouter des éléments à ce modèle	Modèle non vide	Oui	
Cliquer sur le choix <i>DataFlow</i> au dessus du schéma	L'éditeur passe en mode <i>DataFlow</i>	Oui	
Ajouter des éléments aux schéma	<i>DataFlow</i> non vide	Oui	
Cliquer sur le choix <i>Clocks</i> au dessus du schéma	L'éditeur passe en mode <i>Clocks</i>	Oui	
Ajouter des éléments aux schéma	<i>Clocks</i> non vide	Oui	

Scénario C: installation des exemples

Remarques : à partir de la version 0.4.0 d'OpenEmbeDD

Action	Résultat attendu	OK	Commentaires
Dans le Package Explorer : - clic droit puis « New » - « Example... » - choisir le dossier «Polychrony Examples » puis « Polychrony Examples » - clic sur « Finish »	Les projets <i>fr.irisa.espresso.sme.examples</i> doit être créé.	Oui	
Dans le Package Explorer, développer le projet <i>fr.irisa.espresso.sme.exam ples</i> et double cliquer sur les fichiers *.sme pour afficher les modèles.	Ils doivent tous s'ouvrir sans erreur.	Oui	
Double cliquer sur les fichiers *.smedi pour afficher les diagrammes.	Ils doivent tous s'ouvrir sans erreur.	Oui	
Sur chacun d'entre eux, cliquer sur le bouton « <i>Validate the model</i> ».	Tous les modèles doivent être valides.	Oui	

Scénario D: appel des fonctions du compilateur

Remarques : à partir de la version 0.4.0 d'OpenEmbeDD. Se base sur les exemples de modèle installés lors du scénario précédent.

Action	Résultat attendu	OK	Commentaires
Cliquer avec le bouton droit sur SimpleHierarchicAutomaton.sme	Il doit y avoir un menu Polychrony.	Oui	
Sélectionner dans le menu Polychrony, «Generate SIG File».	Un répertoire SimpleHierarchicAutomaton contenant un fichier SimpleHierarchicAutomaton_LIS.SIG a été créé. Dans la console, un message indiquant: ===> Reduction to the kernel language ===> Graph generation (Process SimpleHierarchicAutomaton) # Annotated source program generation: SimpleHierarchicAutomaton_LIS.SIG	Oui	Erreur de scénario. Il faut utiliser Generate SIG (LIS) File.
Ouvrir le fichier SimpleHierarchicAutomaton_LIS.SIG	Le fichier contient un programme Signal en version textuel.	Oui	A noter que dans cette version, il y a un éditeur textuel qui fournit une coloration syntaxique.
Cliquer avec le bouton droit sur Counter.sme et sélectionner dans le menu Polychrony, «Generate C files».	Une boîte de dialogue s'ouvre demandant si on veut forcer la génération de code C. Dans la console, des messages apparaissent annonçant plusieurs fois ===> Clock calculus (Process: Counter)	Oui	
Répondre « Yes »	Un répertoire Counter a été ajouté dans le répertoire model. Il doit contenir: - Counter_body.c - Counter_body.h - Counter externals.h - Counter externalsProc.h - Counter_io.c - Counter_main.c - Counter_types.h Dans la console, un message s'est affiché pour annoncer la génération de ces fichiers	Oui	
Cliquer avec le bouton	Un répertoire Watchdog a	Oui	Erreur de scénario. Les

<p>droit sur Watchdog.sme et sélectionner dans le menu Polychrony, « Generate Java files ».</p>	<p>été ajouté dans le répertoire model. Il doit contenir:</p> <ul style="list-style-type: none">- Watchdog_io.java- Watchdog_main.java- Watchdog.java	<p>nouvelles fonctionnalités de SME font que la génération Java crée un projet Eclipse Java dans lequel les fichiers Java sont générés en ajoutant les libraries Signal Java.</p>
---	---	---

Scénario E : aide en ligne

Remarques : --

Action	Résultat attendu	OK	Commentaires
Dans la barre de menu Eclipse, cliquer sur « Help » puis « Help Contents »	Un item « Polychrony » doit apparaître	Oui	Il s'appelle SME User Manual.
Sélectionner l'item « Polychrony » et le déplier	Des rubriques et/ou une première page de documentation doivent apparaître	Oui	
Passer en revue chaque rubrique et page de documentation	Les pages ne doivent pas comporter d'images absentes (marquées par un cadre vide) et les liens doivent être fonctionnels	Non	Problème dans les liens de la toc qui ne pointent pas au bons endroits. Par contre, en passant par les liens des pages elles-mêmes, c'est OK.