



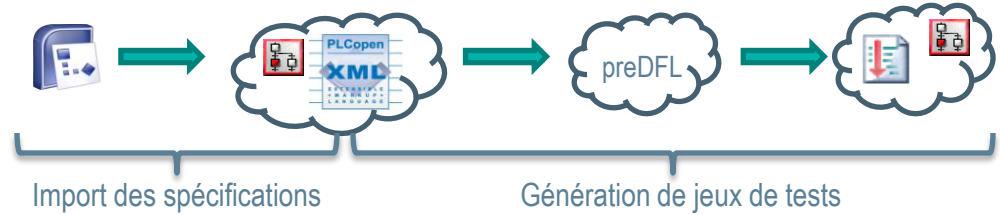
3DEXPERIENCE

Réunion plénière VACSIM

Extension d'algorithmes de test

Christophe Jaouen / Eric Mével
16 avril 2013

Contexte



► Spécification définit

- ▷ un algorithme de positionnement du système dans un état stabilisé (BTP), basé sur un algorithme existant (BT)
- ▷ un algorithme de génération de séquences de tests (ATP)
- ▷ un algorithme de TPPP optimisé
- ▷ des adaptations sur les vérifications en boîte blanche

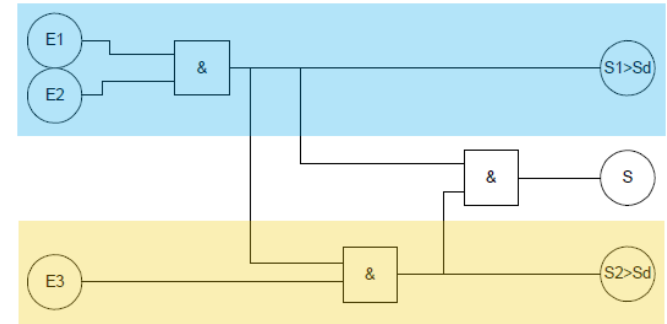
► Objectif : réduire le nombre de jeux de tests en se basant sur les variables intermédiaires

▷ Objectif :

- ▶ $S = f(S1, S2)$
- ▶ $S1 = f(E1, E2)$
- ▶ $S2 = f(S1, E3)$

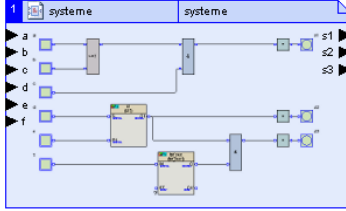
▷ Au lieu de :

- ▶ $S = f(E1, E2, E3)$
- ▶ $S1 = f(E1, E2)$
- ▶ $S2 = f(E1, E2, E3)$



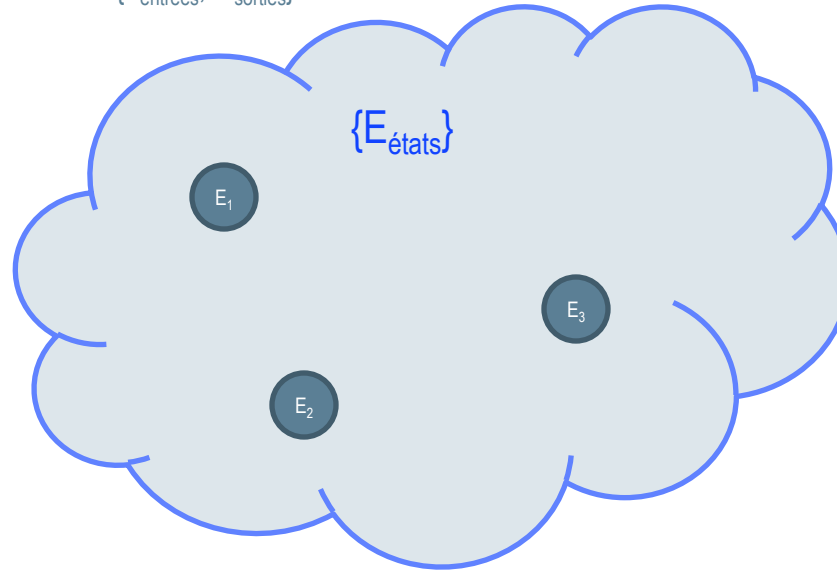
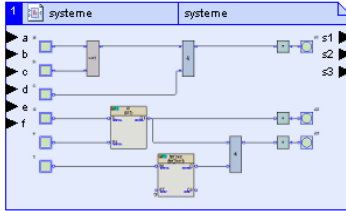
Rappel des algorithmes

Soit un système en boîte blanche possédant $\{n_{\text{entrées}}, m_{\text{sorties}}\}$



Rappel des algorithmes

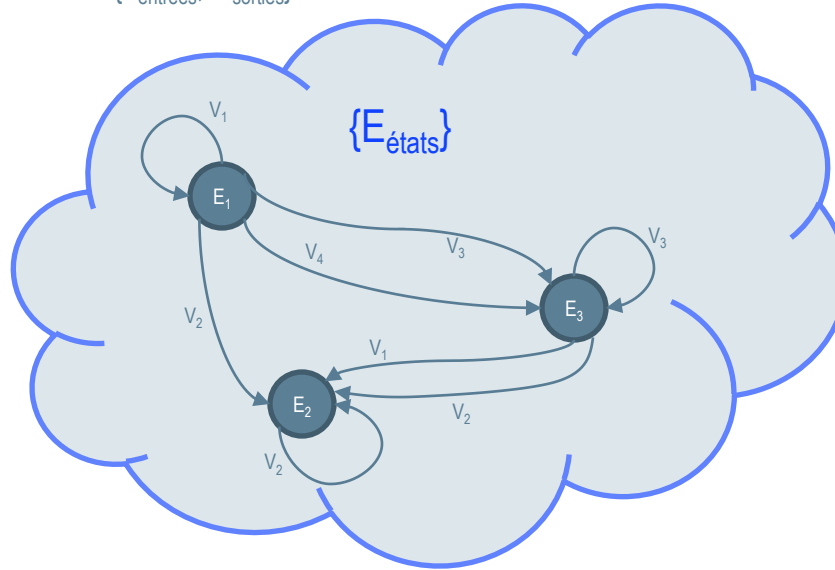
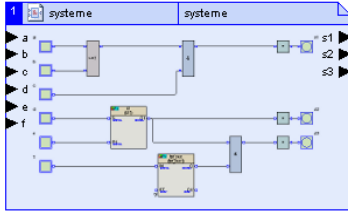
Soit un système en boîte blanche possédant $\{n_{\text{entrées}}, m_{\text{sorties}}\}$



L'algorithme BTP détermine les états stables du systèmes et leur séquence d'atteinte

Rappel des algorithmes

Soit un système en boîte blanche possédant $\{n_{\text{entrées}}, m_{\text{sorties}}\}$

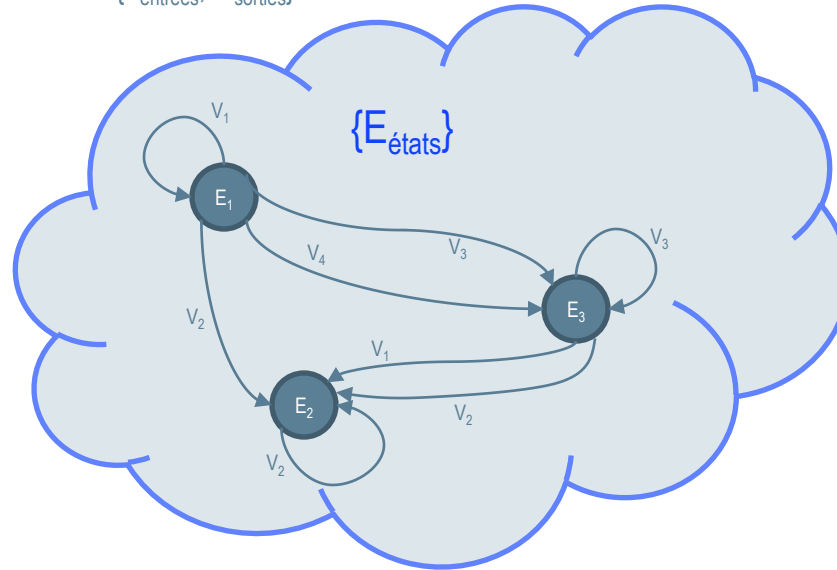
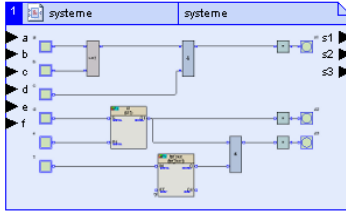


L'algorithme BTP détermine les états stables du systèmes et leur séquence d'atteinte

L'algorithme ATP détermine l'évolution temporelle du système et sélectionne les vecteurs optimums

Rappel des algorithmes

Soit un système en boîte blanche possédant $\{n_{\text{entrées}}, m_{\text{sorties}}\}$



E_1	: $V_1, V_2(E_2), V_2$
E_2	: $V_3(E_3), V_3, V_1(E_2)$
E_3	: $V_4(E_3), V_2(E_2)$

L'algorithme BTP détermine les états stables du système et leur séquence d'atteinte

L'algorithme ATP détermine l'évolution temporelle du système et sélectionne les vecteurs optimaux

L'algorithme TPPP organise les vecteurs sélectionnés par l'algorithme ATP

Etat d'avancement

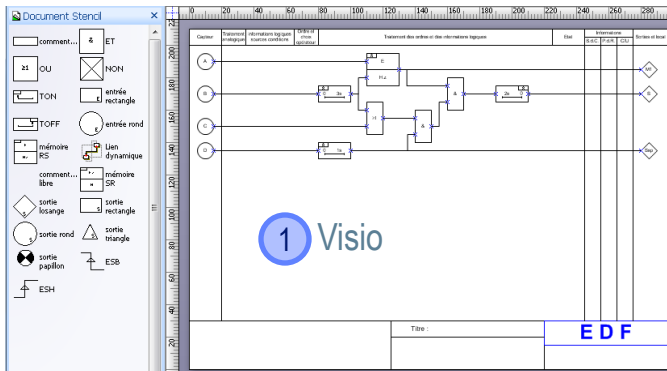
- ▶ Algorithme BTP : ~~codé (non validé)~~
 - ▷ codé, validé sur la base des exemples donnés dans la spécification
- ▶ Algorithme ATP : ~~en cours~~
 - ▷ codé, validé sur la base des exemples donnés dans la spécification
- ▶ Algorithme TPPP optimisé : ~~à venir~~
 - ▷ codé, excepté les stratégies introduites début février
- ▶ Modification des vérifications en boîte blanche : ~~à venir~~
 - ▷ à venir

Démo

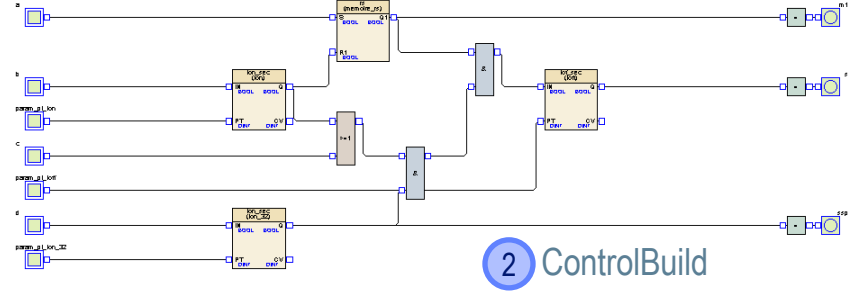
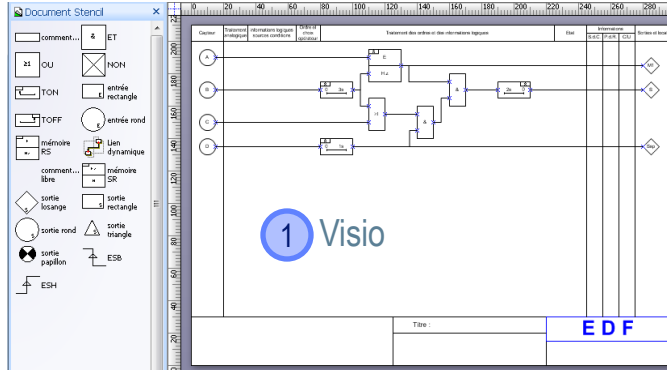


3DEXPERIENCE

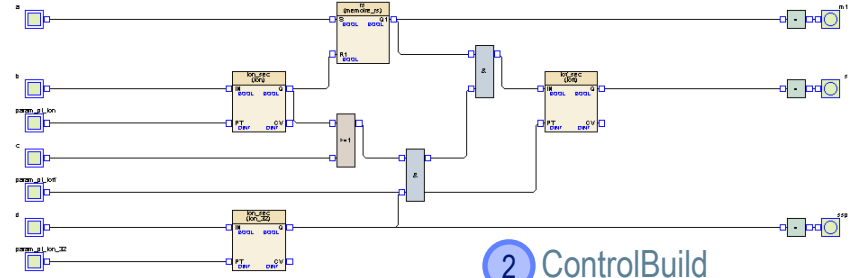
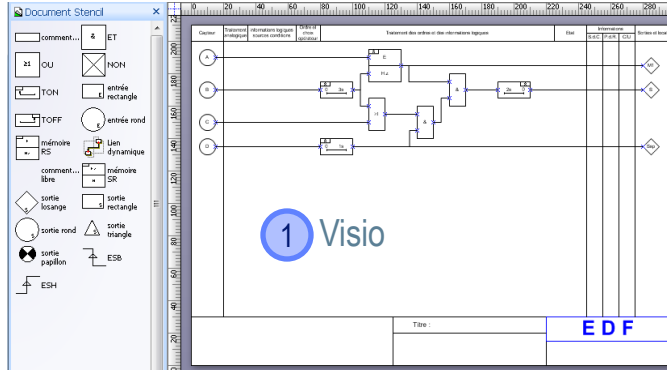
Procédure



Procédure



Procédure

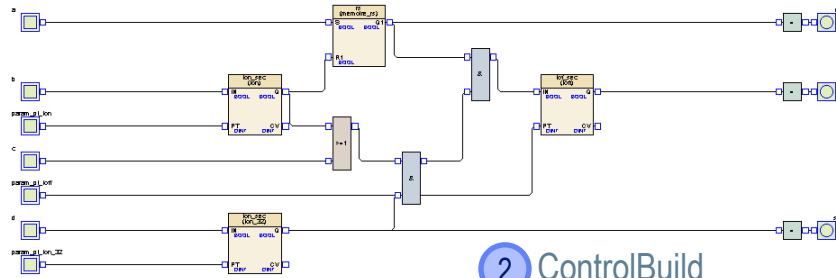
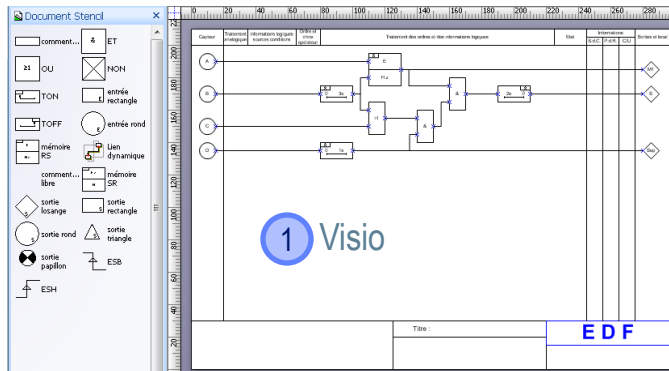


3 preDFL

```

SSP=TON (TON_32)
S=TOFF (TOFF)
TOFF (TOFF, 2xS, ET (RS (MEMOIRE_RS_Q1), OU (TON (TON), E (C)), TON (TON_32)))
TON (TON_32, 1xS, E (D))
M1=RS (MEMOIRE_RS_Q1)
RS (MEMOIRE_RS_Q1, E (A), TON (TON))
TON (TON, 3xS, E (B))
    
```

Procédure



3 preDFL

Séquence de tests suite à l'application de l'algorithme TPPP optimisé sur la sortie S

Etat de départ

A:0 B:1 C:0 D:0

MEMOIRE_RS_Q1:0 TOFF:A TON:A TON_32:D SSP:0

Sortie = 0

Vecteur appliqué

A:0 B:1 C:0 D:0

Etats réduits atteints

MEMOIRE_RS_Q1:0 TOFF:A TON:A TON_32:D SSP:0

Sortie = 0

Vecteur appliqué

A:0 B:1 C:1 D:0

Etats réduits atteints

MEMOIRE_RS_Q1:0 TOFF:A TON:A TON_32:D SSP:0

Sortie = 0

4 Fichier TPPP



```
SSP=TON (TON_32)
S=TOFF (TOFF)
TOFF (TOFF, 2xS, ET (RS (MEMOIRE_RS_Q1), OU (TON (TON), E (C)), TON (TON_32)))
TON (TON_32, 1xS, E (D))
M1=RS (MEMOIRE_RS_Q1)
RS (MEMOIRE_RS_Q1, E (A), TON (TON))
TON (TON, 3xS, E (B))
```

Résultats obtenus

Les résultats partiels obtenus ont permis

- ▷ de faire évoluer la spécification, notamment, d'affiner les exemples présentés
- ▷ d'approfondir la réflexion quant à la séquence de tests à retenir (objet de l'évolution de la spécification de L1.1V7 à L1.1V8)

Stratégies possibles pour le TPPP

- ▶ L'algorithme ATP sélectionne les vecteurs intéressants et leur attribue un type
 - ▷ Set, Reset ou Inchangé
- ▶ Organisation des vecteurs de test selon une séquence RII...IISII...IIRSRS
- ▶ Problématique quand nombre différent de vecteurs Set et Reset
 - ▷ Soit la séquence joue tous les vecteurs restants du même type
 - ▷ Soit la séquence continue en alternant les vecteurs Set et Reset en rejouant des vecteurs déjà joués d'un des types

Prochaines étapes

- ▶ Prototypage des 2 stratégies définies pour évaluation de celles-ci
- ▶ Reprise de l'intégration dans l'environnement ControlBuild
 - ▷ Homogénéisation des langages de programmation
 - ▷ Suppression des fichiers intermédiaires (preDLF et TPPP)
- ▶ Génération des scénarios de tests pour ControlBuild

