Introduction aux tests du logiciel

F.X. Fornari
xavier.fornari@esterel-technologies.com
P. Manoury
pascal.manoury@pps.univ-paris-diderot.fr

2014

Des jeux de tests : pour quoi faire ?

- Trace pour le contrôle (le vérificateur vérifie que le testeur a effectué les tests du jeu de test),

Des jeux de tests : pour quoi faire ?

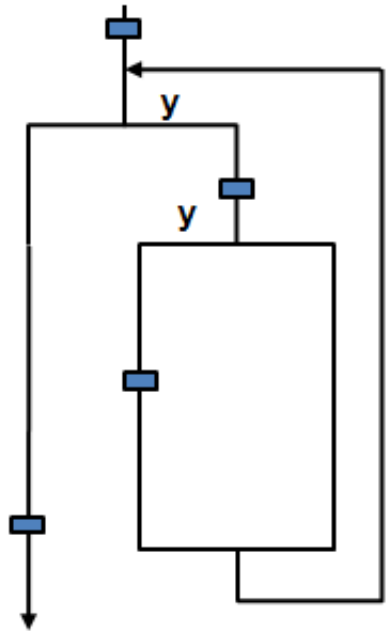
- Trace pour le contrôle (le vérificateur vérifie que le testeur a effectué les tests du jeu de test),
- Trace entre la conception et l'exécution du test (entre le moment de la spécification et celui o'u le code est écrit et peut être testé)

Des jeux de tests : pour quoi faire ?

- Trace pour le contrôle (le vérificateur vérifie que le testeur a effectué les tests du jeu de test),
- Trace entre la conception et l'exécution du test (entre le moment de la spécification et celui o'u le code est écrit et peut être testé)
- Séparer l'exécution des tests de l'analyse des résultats (ce n'est pas au moment o'u on fait le test qu'il faut juger de la pertinence des sorties, ce n'est pas la même personne qui le fait la plupart du temps)

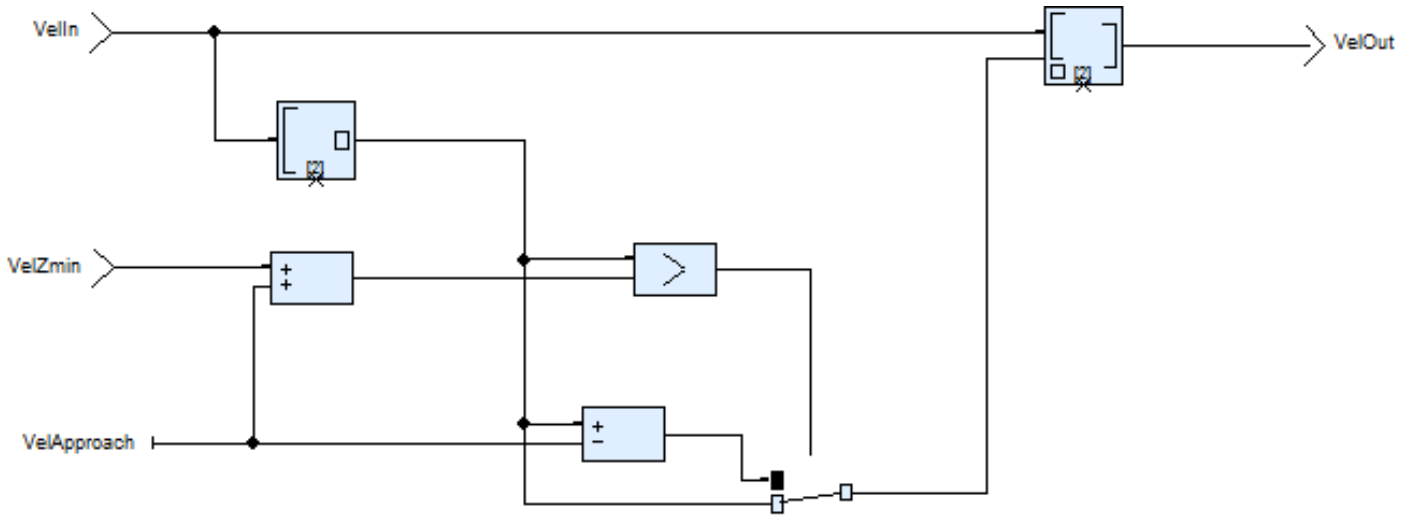
Flot de contrôle

```
Found  = False;
Indice = -1;
while (! Found && Indice < TabMax ) {
  Indice++;
  if ( Tab[Indice] == Value ) {
    Found = True;
  }
}
return Indice;
```



L'analyse du flot de contrôle permet la construction des lignes du tableau de tests

Flot de données



L'analyse du flot de données permet la construction des colonnes du tableau de tests

Composant: exemple

```
VG1, VG2 : Int;
procedure Proc (P1: in Int;
                P2: in out Int;
                P3: out Int)
is
  L1, L2 : Int;
begin
  -- corps de la procédure
end Proc;
```

Composant

Un composant (c'est-à-dire une procédure ou une fonction, *unit* en anglais) peut être caractérisé par :

- ses entrées :
  - paramètres d'entrée du composant
  - variables globales lues dans le composant
- ses sorties :
  - paramètres de sortie du composant
  - variables globales écrites dans le composant

Composant

Un composant (c'est-à-dire une procédure ou une fonction, *unit* en anglais) peut être caractérisé par :

- ses entrées :
  - paramètres d'entrée du composant
  - variables globales lues dans le composant
- ses sorties :
  - paramètres de sortie du composant
  - variables globales écrites dans le composant

Identifiables soit syntaxiquement, soit par une analyse du code

- les relations entre les entrées et les sorties de la procédure (corps de la procédure)
- les variables lues/écrites par les composantes appelées (causes d'indétermination si elles ne sont pas initialisées, cf bouchonnage)

Jeux de test

	Entrées du jeu			Sorties du jeu			
	P1	P2	VG1	P2	P3	VG1	VG2
jeu 1							
jeu 2							
jeu 3							

En supposant que les variables globales *VG1* et *VG2* soient:

- *VG1* lue et écrite
- *VG2* seulement écrite

L'Oracle

Procédure qui permet de prédire la valeur des sorties par rapport aux valeurs d'entrées.

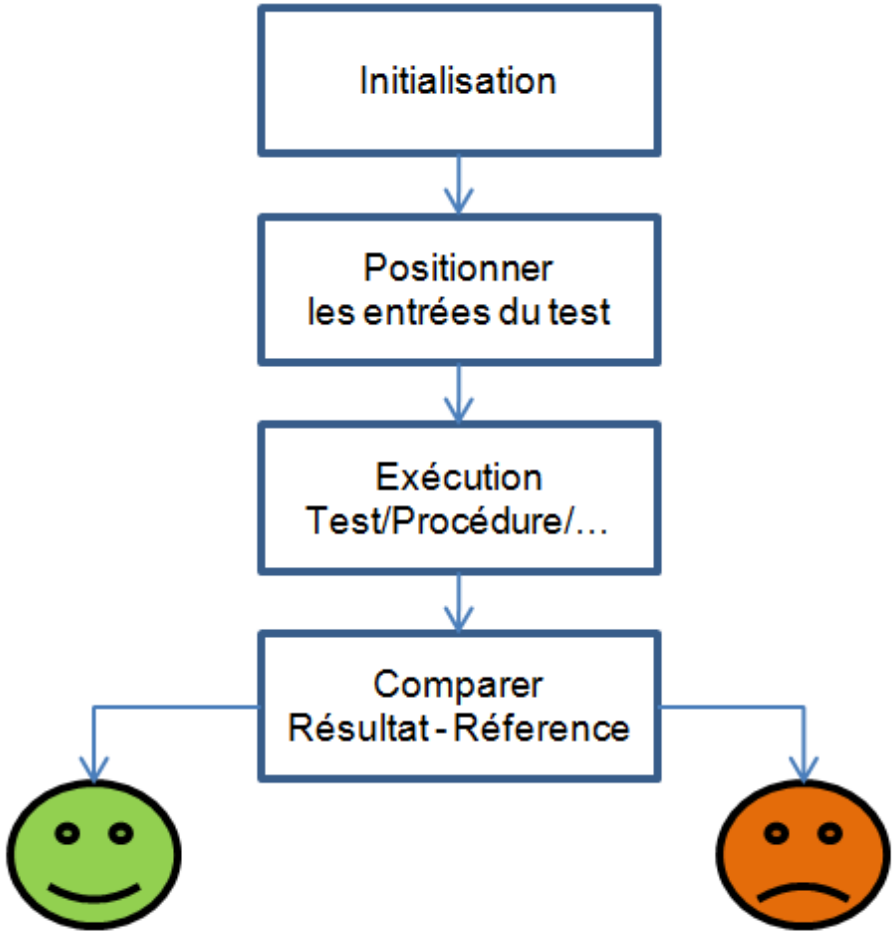
- Bas niveau : valeur précise
- Haut niveau : de la valeur précise à une simple propriété à vérifier (intervalle, propriété mathématique, propriété de sûreté, etc.)

Difficulté du test d'un logiciel

Toute la difficulté du test d'un logiciel provient de la détermination :

- d'un jeu d'entrées par rapport à une couverture de tests
- des valeurs de sortie d'une procédure par rapport à un jeu d'entrées déterminé (problème de l'oracle)
- traitement des composants appelés (cf le bouchonnage)

Exécution d'un test



Importance du jeu de tests

On voit ici l'importance du jeu de tests, car tout repose sur lui :

- Le jeu de tests est une représentation pour des valeurs particulières de la spécification d'une procédure.

Importance du jeu de tests

On voit ici l'importance du jeu de tests, car tout repose sur lui :

- Le jeu de tests est une représentation pour des valeurs particulières de la spécification d'une procédure.
- Si l'on veut tester complètement un logiciel, il faut élaborer tous les jeux de tests permettant de couvrir la spécification.

Importance du jeu de tests

On voit ici l'importance du jeu de tests, car tout repose sur lui :

- Le jeu de tests est une représentation pour des valeurs particulières de la spécification d'une procédure.
- Si l'on veut tester complètement un logiciel, il faut élaborer tous les jeux de tests permettant de couvrir la spécification.
- La combinatoire de n'importe quel problème même de très petite taille est trop importante pour faire un test exhaustif. Par exemple, pour vérifier l'addition sur les entiers 32 bits, cela nécessiterait 2^{64} jeux de tests.

Equilibre d'arbitrage OK/KO

- trop de OK : le client n'est pas content
- trop de KO : le développeur est mécontent (30 à 50% des KO sont liés à des erreurs du testeur)

Il faut séparer (au moins temporellement) l'arbitrage et l'exécution des tests.

Couverture de tests, critère d'arrêt

Couverture de tests

- niveau de confiance dans le logiciel pour le client,
- le contrat entre le client et le testeur, jugé par le vérificateur,
- pour le testeur, critère de mesure

Couverture de tests, critère d'arrêt

Couverture de tests

- niveau de confiance dans le logiciel pour le client,
- le contrat entre le client et le testeur, jugé par le vérificateur,
- pour le testeur, critère de mesure

Critère d'arrêt : quand s'arrêter de tester ?

- négatif : bugs bloquants, on n'est pas en mesure de tester la suite
- positif (taux de couverture)

On ne cherche pas 100% de la couverture à chaque fois ($> 90\%$ bien fait, sauf normes de sûreté très spécifiques)

Choix de la couverture de tests

3 critères de choix :

- criticité du logiciel (normes de sûreté, imposé par le vérficateur)
- contraintes imposées au logiciel (facteurs qualité imposés par le client : temporelles, portabilité, etc.)
- type de logiciel (domaine : BD, réseaux, embarqué, etc.)

Choix de la couverture de tests

3 critères de choix :

- criticité du logiciel (normes de sûreté, imposé par le vérficateur)
- contraintes imposées au logiciel (facteurs qualité imposés par le client : temporelles, portabilité, etc.)
- type de logiciel (domaine : BD, réseaux, embarqué, etc.)

Taux de couverture

C'est une mesure de la couverture effective des tests.
Des justifications peuvent être nécessaires.

Choix de la couverture de tests

3 critères de choix :

- criticité du logiciel (normes de sûreté, imposé par le vérficateur)
- contraintes imposées au logiciel (facteurs qualité imposés par le client : temporelles, portabilité, etc.)
- type de logiciel (domaine : BD, réseaux, embarqué, etc.)

Taux de couverture

C'est une mesure de la couverture effective des tests.
Des justifications peuvent être nécessaires.

Chaque jeu de test doit augmenter la couverture de tests