

Candide au pays de FoCa1

P.Ayrault

8 décembre 2006

Introduction

Présentation du voteur

Architecture du voteur

- Décomposition organique

- Le voteur

- Les valeurs

- Exécution

Principales difficultés rencontrées

Conclusions

Perspectives

- ▶ Prise en main de l'atelier FoCa1
- ▶ Faisabilité d'utilisation de l'atelier FoCa1 dans le domaine de la Sûreté de Fonctionnement (SdF)
- ▶ Identification d'améliorations potentielles dans les constructions du langage FoCa1 spécifique pour la SdF

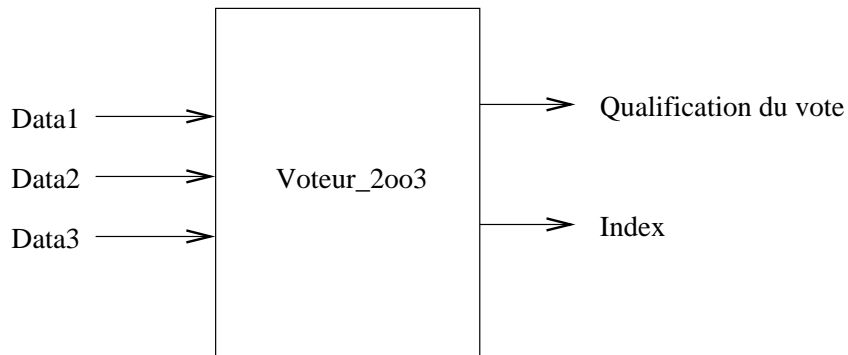
¹Ce travail s'inscrit dans le cadre de l'ANR SSURF (**S**afety and **S**ecurity **U**nder **R** Focal) ainsi que d'une thèse sur la réalisation d'études de sécurité de type dysfonctionnel à partir d'un modèle FoCa1 .

La redondance des différents éléments d'un système est une technique la plus fréquemment utilisées pour se prémunir contre les défaillances aléatoires

Nécessité de voter les données des différentes répliques pour prendre une décision

Plusieurs type de redondance

- ▶ 1oo2 (i.e. 1 parmi 2), permet d'obtenir une grande disponibilité
- ▶ 2oo2 (i.e. 2 parmi 2), permet d'obtenir une grande sécurité
- ▶ 2oo3 (i.e. 2 parmi 3), permet d'obtenir une grande sécurité et une disponibilité accrue



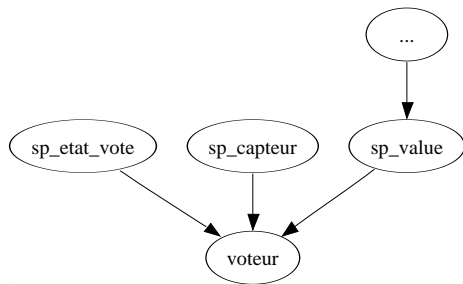
- ▶ **perfect_match** les 3 valeurs en entrées sont cohérentes, un des index est retourné
- ▶ **partial_match** 2 des 3 valeurs en entrée sont cohérentes entre elles et la troisième est inconsistante, l'index de la valeur inconsistante est retournée
- ▶ **range_match** 1 valeur est cohérente avec les 2 autres qui sont mutuellement inconsistante, l'index de cette valeur est retournée
- ▶ **no_match** toutes les valeurs sont inconsistantes entre elles. Un des index est retourné.

Transition ?? Architecture du voteur

- ▶ **Espèce de spécification** nommée *sp_*xxx contient
 - ▶ la déclaration des fonctions *exportées* de l'espèce
 - ▶ la description fonctionnelle de la fonction sous forme de propriétés
 - ▶ les propriétés portant sur les fonctions
 - ▶ la preuve des propriétés sur les fonctions
 - ▶ les contraintes sur les paramètres sous forme de propriétés
- ▶ **Espèce d'implémentation** nommée *imp_*xxx contient
 - ▶ la définition des fonctions
 - ▶ la preuve des descriptions fonctionnelles
- ▶ **Collection**
 - ▶ association des différentes implémentations complètes
 - ▶ preuve des contraintes sur les paramètres

Le lien entre les espèces de spécification et les espèces d'implémentation est réalisé par héritage.

Décomposition organique

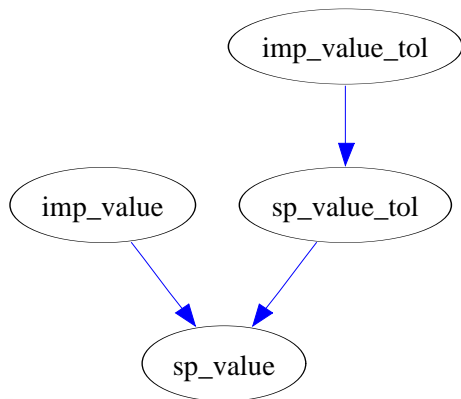


Espèce de spécification *Voteur*

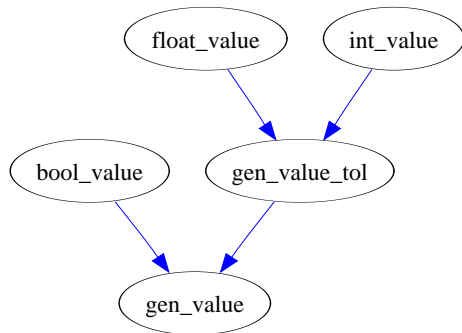
```
(** Le voteur 2003 *)
species sp_voteur(e is sp_etat_vote , c is sp_captteur ,
  inherits basic_object =
rep = c * e ;

sig vote in v -> v -> v -> self ;

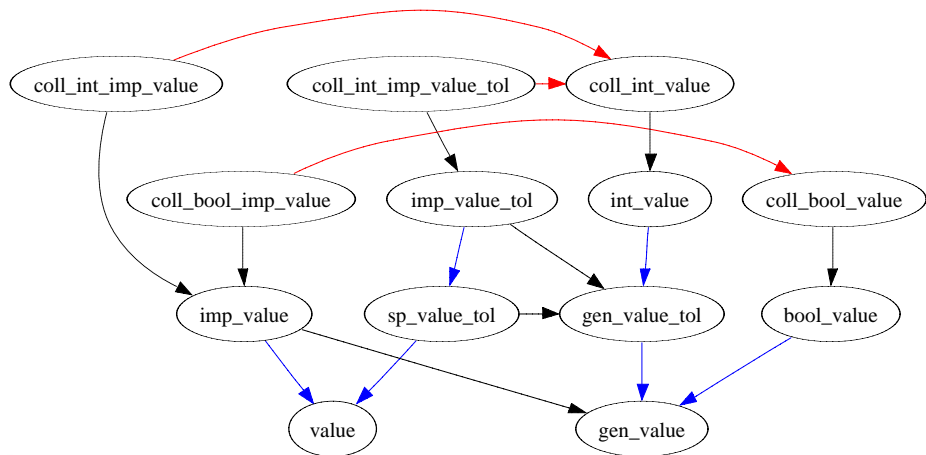
(* Vote avec 3 valeurs coherentes *)
property vote_perfect :
  all v1 v2 v3 in v ,
    ((v!valeur_coherente(v1 , v2) and
      v!valeur_coherente(v2 , v3) and
      v!valeur_coherente(v1 , v3))
  ->
    (c!equal(!captteur(!vote( v1 , v2 , v3)), c!capt_1) and
      e!equal(!etat(!vote( v1 , v2 , v3)), e!perfect_matchl
...
(* Proprietes sur la coherence de valeur *)
property equal_value_is_symmetric :
  all v1 v2 in v
```



Les types supports



Les collections



Voteur entier avec tolerance de 2

v1 : 1, v2 : 3, v3 : 5 --> val : capt_2 , res : partial_match

v1 : 1, v2 : 1, v3 : 5 --> val : capt_3 , res : range_match

v1 : 4, v2 : 5, v3 : 5 --> val : capt_1 , res : perfect_match

v1 : 1, v2 : 4, v3 : 7 --> val : capt_1 , res : no_match

Voteur entier sans tolerance

v10 : 1, v20 : 3, v30 : 5 --> val : capt_1 , res : no_match

v1 : 5, v2 : 5, v3 : 5 --> val : capt_1 , res : perfect_match

v1 : 4, v2 : 5, v3 : 5 --> val : capt_1 , res : range_match

v1 : 1, v2 : 4, v3 : 7 --> val : capt_1 , res : no_match

Voteur boolean sans tolerance

v1 : False, v2 : False, v3 : True --> val : capt_3 , res : partial_match

v1 : False, v2 : False, v3 : False --> val : capt_1 , res : no_match

v1 : False, v2 : False, v3 : True --> val : capt_3 , res : partial_match

v1 : False, v2 : False, v3 : False --> val : capt_1 , res : no_match

Transition ?? Principales difficultés rencontrées

Difficulté : Obligation d'implémenter toutes les définitions de l'espèce de spécification

- ▶ Une spécification permet de déclarer une espèce sans privilégier une implémentation particulière
- ▶ Une implémentation a pour objectif de fournir une espèce la plus performante (optimisée) possible respectant la spécification
- ▶ Introduction de la notion de *concept*. Une déclaration nécessaire uniquement dans une espèce de spécification mais non définie dans les espèces d'implémentation (pour cause d'optimisation)

Difficulté : Self doit obligatoirement être monobloc.

- ▶ Les espèces manipulées contiennent généralement plusieurs attributs
- ▶ Obligation de définir (puis d'implémenter) des fonctions de projection pour chaque attribut de la rep
- ▶ Impossible de fournir une description fonctionnelle des fonctions de projection, donc pas de preuve possible sur l'implémentation de ces fonctions

L'utilisation du *in* dans les paramètres d'une espèce

Difficulté : Comportement étrange lors de l'utilisation de l'opérateur *in* dans les paramètres d'une espèce

- ▶ Utilisation d'une espèce avec un paramètre *in* en paramètre d'une espèce
- ▶ `species voteur(t is gen_value_tol, tol in t, v is sp_value(t, tol))`
- ▶ Lors de la création d'une collection, il y a perte de la liaison entre les 2 instances du paramètre dans l'espèce

- ▶ Expression des propriétés portant sur les entiers
- ▶ Pas de possibilité de définir des variables intermédiaires (let .. in) dans l'écriture des preuves
- ▶ Peu d'information retournée par Zenon (prove/fail), par contre puissance de la preuve

- ▶ Faisabilité de la modélisation sur un exemple simple et fortement combinatoire
- ▶ Difficultés pour la description des propriétés et la réalisation des preuves dans le domaine arithmétique
- ▶ Besoins de définition d'une *frontière* entre la spécification et l'architecture

- ▶ Modèle pour la réalisation d'automates
- ▶ *Template d'espèce* pour les études dysfonctionnelles (ANR SSURF)
- ▶ Dataflow sur un modèle FoCa1