IRSN

Journée SEE
"Certification et méthodes formelles"

Utilisation de l'outil GATeL pour l'évaluation des tests de logiciels critiques à l'IRSN

ONERA Toulouse, 3 février 2004

Jean Gassino (jean.gassino@irsn.fr).
Pascal Régnier (pascal.regnier@irsn.fr)

Plan

Contexte

- Réglementaire
- Technique

Couverture des tests fonctionnels

- Besoin de l'IRSN
- Critères
- Outillage
- Résultats

Conclusions et perspectives

"Certification" des logiciels

Acteurs de la sûreté nucléaire en France

Exploitants: EDF, COGEMA, CEA,...

Autorité de sûreté: Direction Générale de la Sûreté Nucléaire et de la Radioprotection + antennes régionales (260 personnes)

Appui technique: Institut de Radioprotection et de Sûreté Nucléaire (expertise et recherche, 1500 personnes)

Démarche

- Pas de "certification" d'équipements ou de logiciel mais des autorisations de fonctionnement pour une installation...

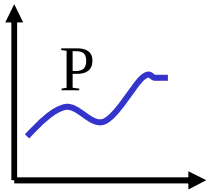
... délivrées sur la base d'un avis IRSN élaboré à l'issu d'un débat technique contradictoire.

Référentiel pour les logiciels critiques

- RFS II.4.1a: Règle fondamentale de sûreté/logiciels critiques
- norme CEI 60880

Illustration du rôle du SYSTEME DE PROTECTION

Contrôle commande non classé de sûreté (régulations)



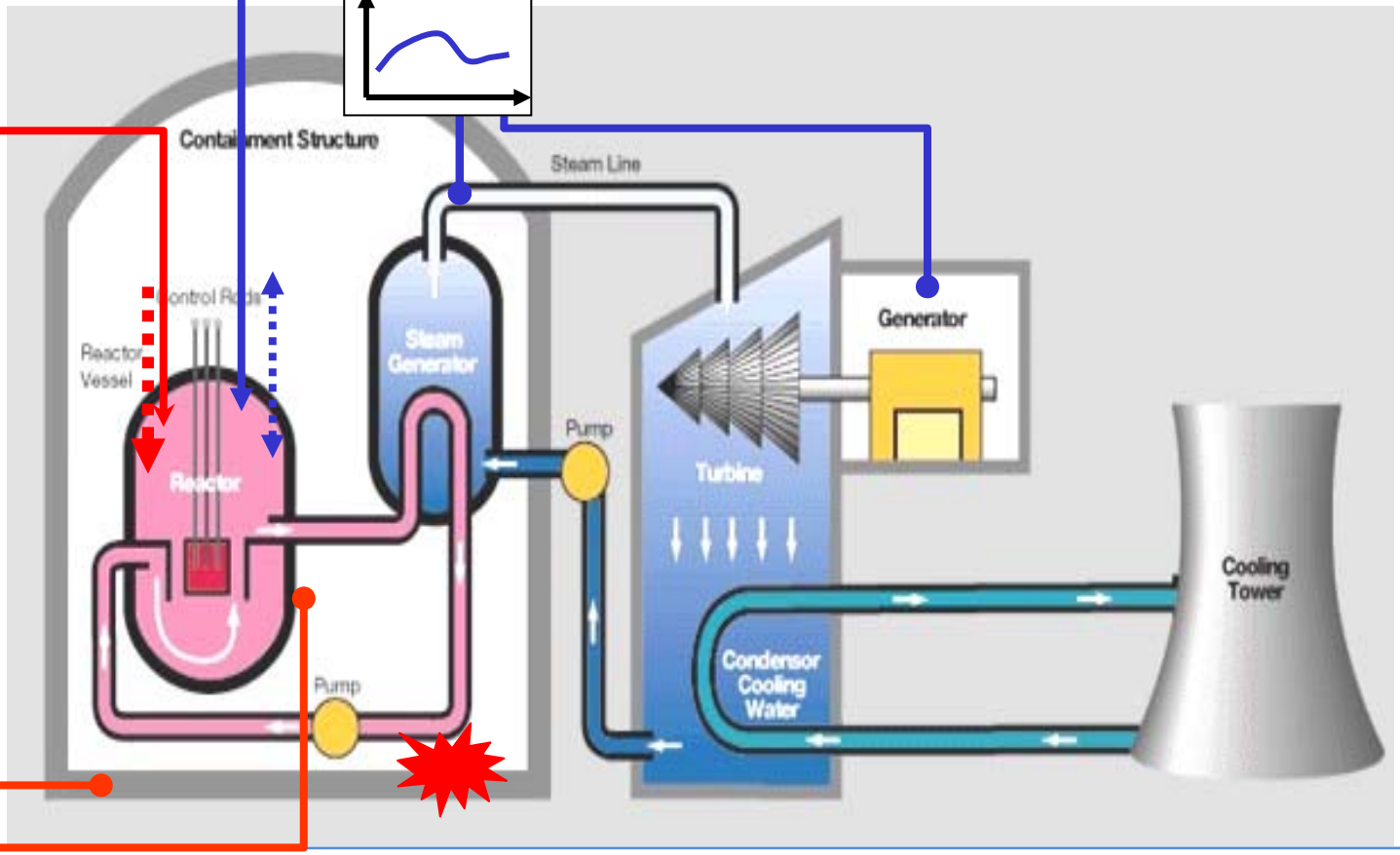
Fonctionnement normal

Incident

Système de protection

Observation « passive » en fonctionnement normal

Agit sur détection d'un événement initiateur



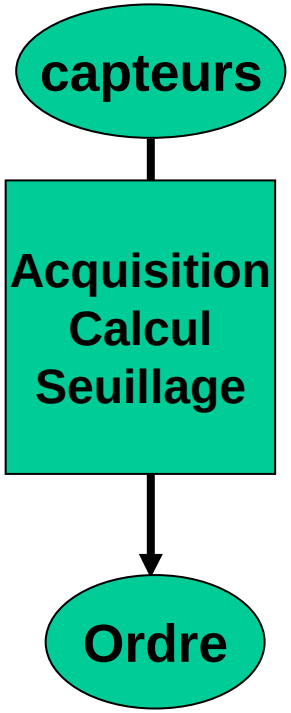
Systèmes et logiciels critiques du nucléaire

Fonctions de sûreté confiées au logiciel:

- études d'accidents -> événements initiateurs -> chaînes de protection
  - ex: ordre de chute des barres (ou démarrage autre actionneur) sur:
    - haut flux neutronique sous condition...
    - puissance thermique élevée sous condition...
    - basse température sous condition...
    - ...
- systèmes principalement en boucle ouverte; en général pas de régulation, mais quelques rebouclages (permissifs et inhibitions)
- systèmes dédiés: tout le logiciel concourt à la fonction de sûreté!
- entrées logiques et analogiques,
- sorties Tout ou Rien (avec souvent une position de repli sûre = déclenchement)

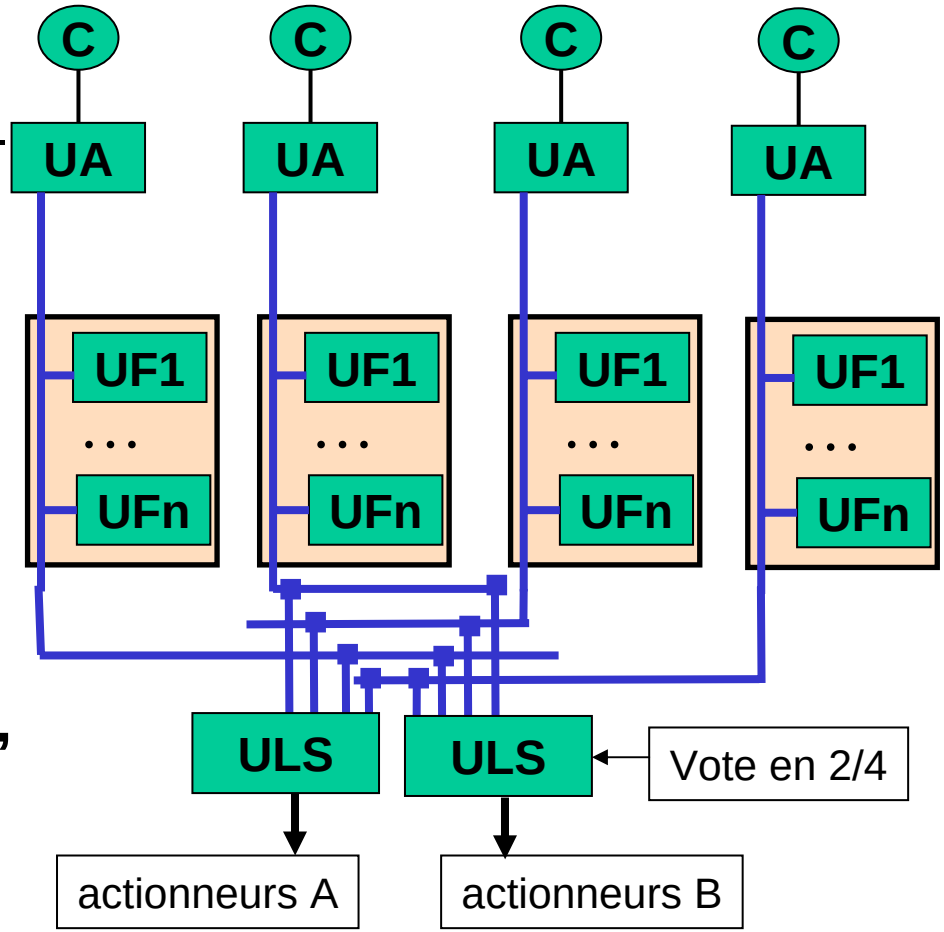
Systèmes et logiciels critiques du nucléaire

Chaîne



Architecture typ. :

- Calculateurs de sûreté dédiés
- Redondance 4
- 3 étages/train
- Réseaux déterministes
- Unités synchrones, asynchronisme global



UA Unité d'Acquisition
UF Unité Fonctionnelle
ULS Unité Logique de Sauvegarde
Réseau

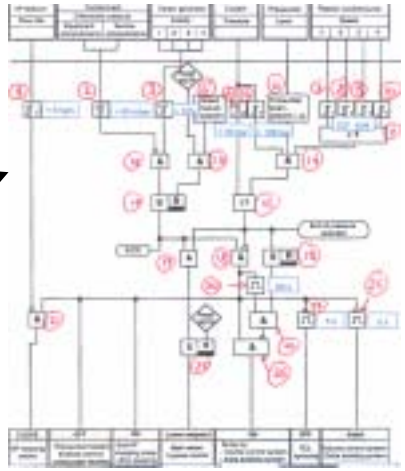
Développement des logiciels

Application

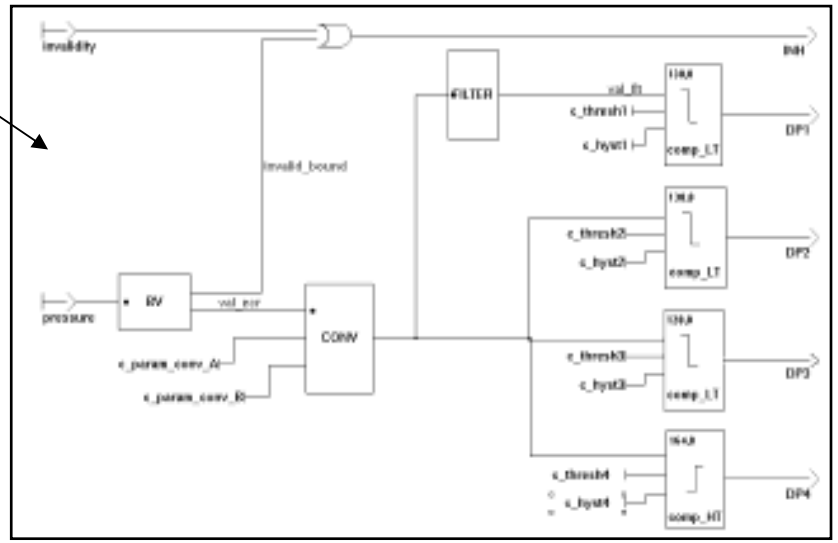
- Spécification:
  - -diagrammes fonctionnels informels
  - -texte complémentaire
- Conception *manuelle* dans un AGL (Scade..)
  - interprétation spec
  - adaptation à la plate-forme
  - langage graphique orienté métier
- Codage:
  - -génération automatique de code (C)

Système

- Logiciel système minimal dédié
- Développement "manuel".



Cf. transparent suivant



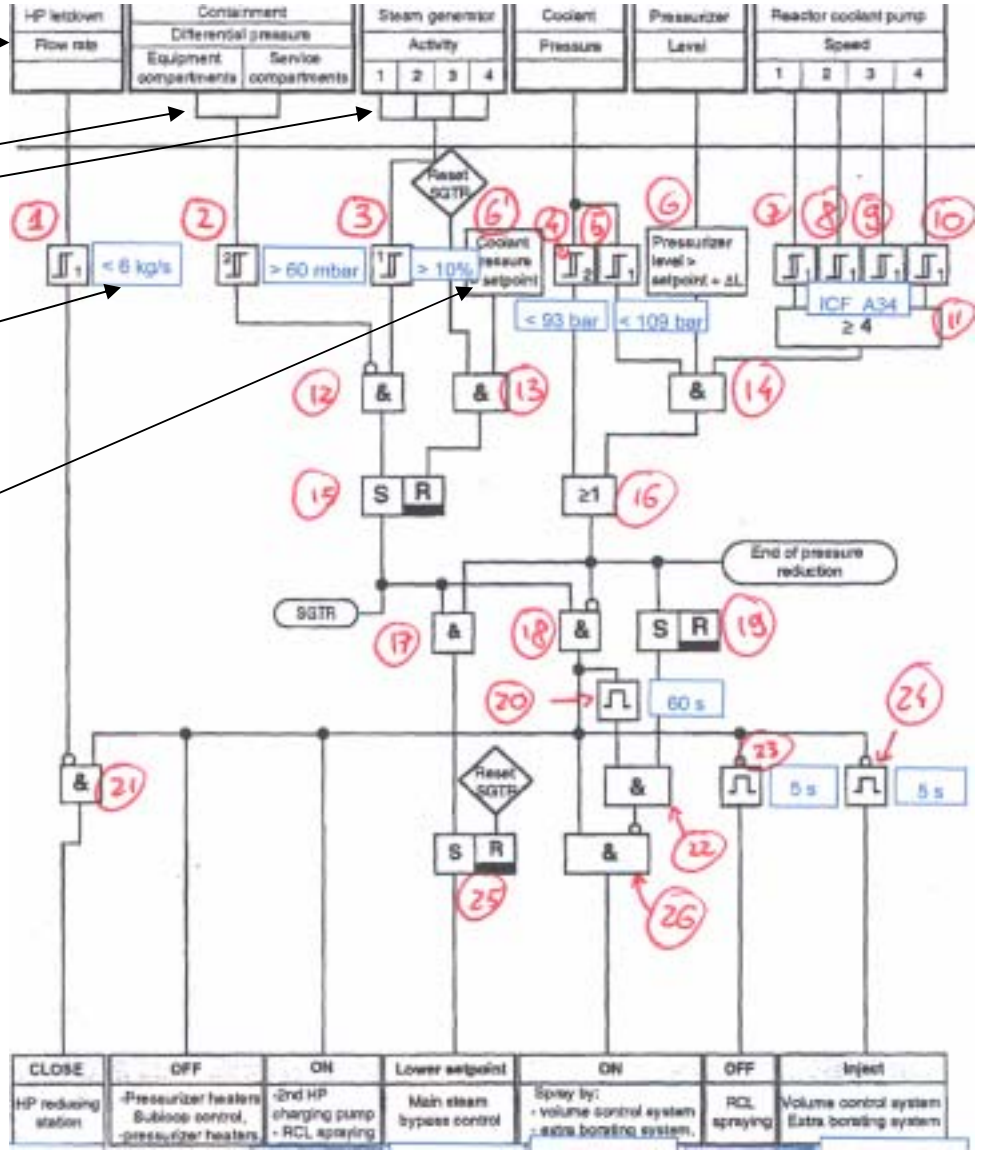
Aspect d'une spécification

entrées abstraites
(asynchronismes..)

N dans 1…

unités fonctionnelles

boîte sans entrée visible



Critères de couverture des tests fonctionnels

Constat: pas de véritable critère de couverture pour les tests fonctionnels

- Pour le test structurel il existe des critères non ambigus pouvant être appliqués par des outils du commerce: couverture des lignes, des instructions, des MCDC,..

- Pour le test fonctionnel le critère se résume souvent à "tester toutes les exigences...."

Que serait un "bon critère" de couverture fonctionnelle?

- applicable à tout logiciel de contrôle-commande de sûreté
- exprimer une couverture des fonctions et non de la structure du code
- l'application par différentes personnes doit conduire au même résultat
-> ... recherche d'une définition formelle.

Nb:
- une couverture, même fonctionnelle se réfère à une structure (couvrir une liste de fonctions, un modèle,...)
- ... il n'y a vraisemblablement pas un bon critère unique ...

Critères de couverture des tests fonctionnels

Proposition de 2 critères

- critère 1: couverture des comportements locaux des modules
- critère 2: couverture des conditions logiques élémentaires

Ces critères :

- sont fondés sur la couverture des spécifications fonctionnelles amont

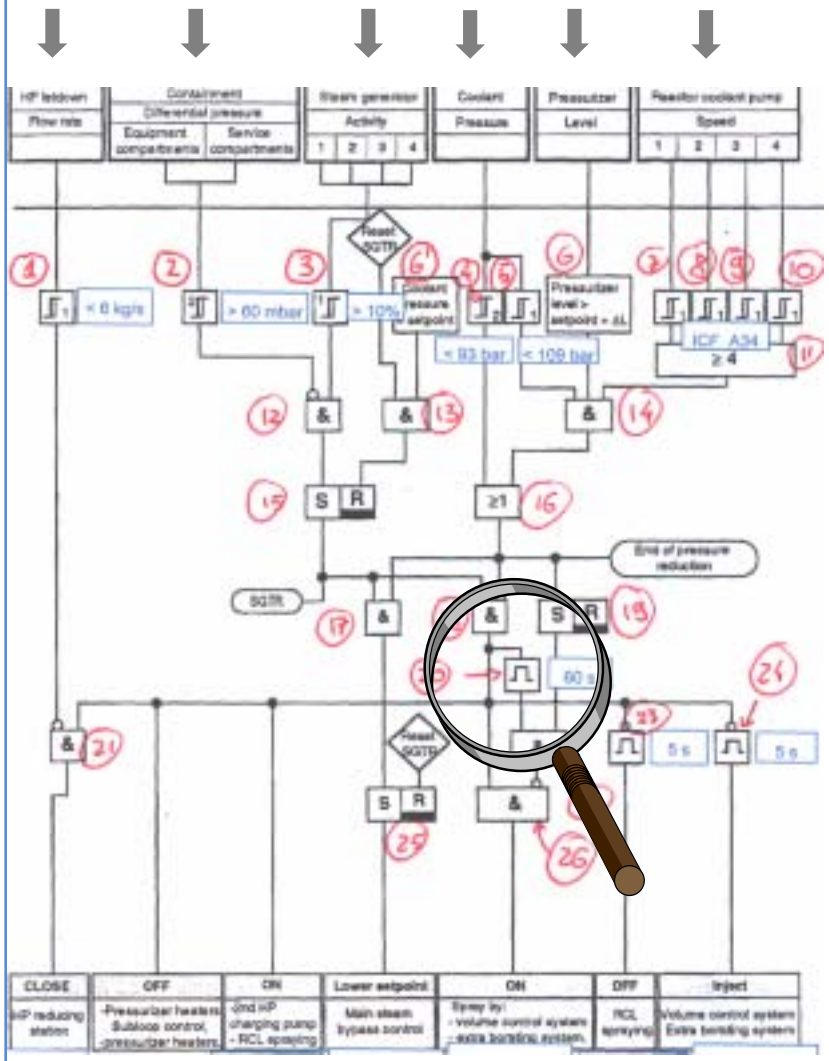
- concernent les tests de validation fonctionnelle du logiciel;

- sont définis et appliqués de manière formelle (langage LUSTRE).

... au delà des 2 critères proposés, le plus important est la démarche et l'existence d'un outil qui permet de définir d'autres critères...

Critères de couverture des tests fonctionnels

1er critère: Couverture des comportements locaux des modules



Principe:

1 définir des catégories de comportement pour chaque type de module.

2 couvrir toutes ces catégories pour chaque module apparaissant dans le diagramme.

Exemples :

- and, or… table de vérité (ou partie de…)

- impulsion

cat 1: sollicitation simple isolée (« normale »)

cat 2: entrée initialement à 1 (condition de démarrage non nominale)

cat 3: nouvelle sollicitation alors que la sortie est encore armée(permet la discrimination module réarmable/module non réarmable)

- bascule

cat 1: …

cat 2:…

cat 3: entrées Set et Reset simultanément à 1 (priorité de R sur S)

Critères de couverture des tests fonctionnels

Remarques concernant ce premier critère

Il ne s'agit pas de test structurel

On observe certes le comportement local du module mais:

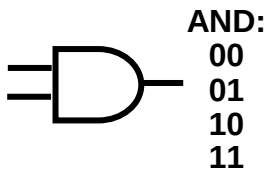
- **plongé dans le logiciel complet (test de validation avec les entrées du logiciel)**

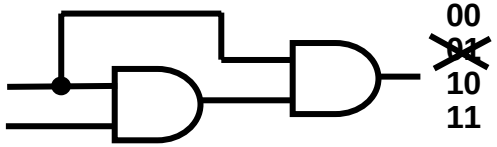
- **les catégories sont définies pour détecter de mauvaises connections ou un mauvais paramétrage des modules, et non leur bonne implémentation**

- **le schéma de départ est informel, ambigu, ce n'est pas l'implémentation**

Il peut exister des catégories non atteignables:

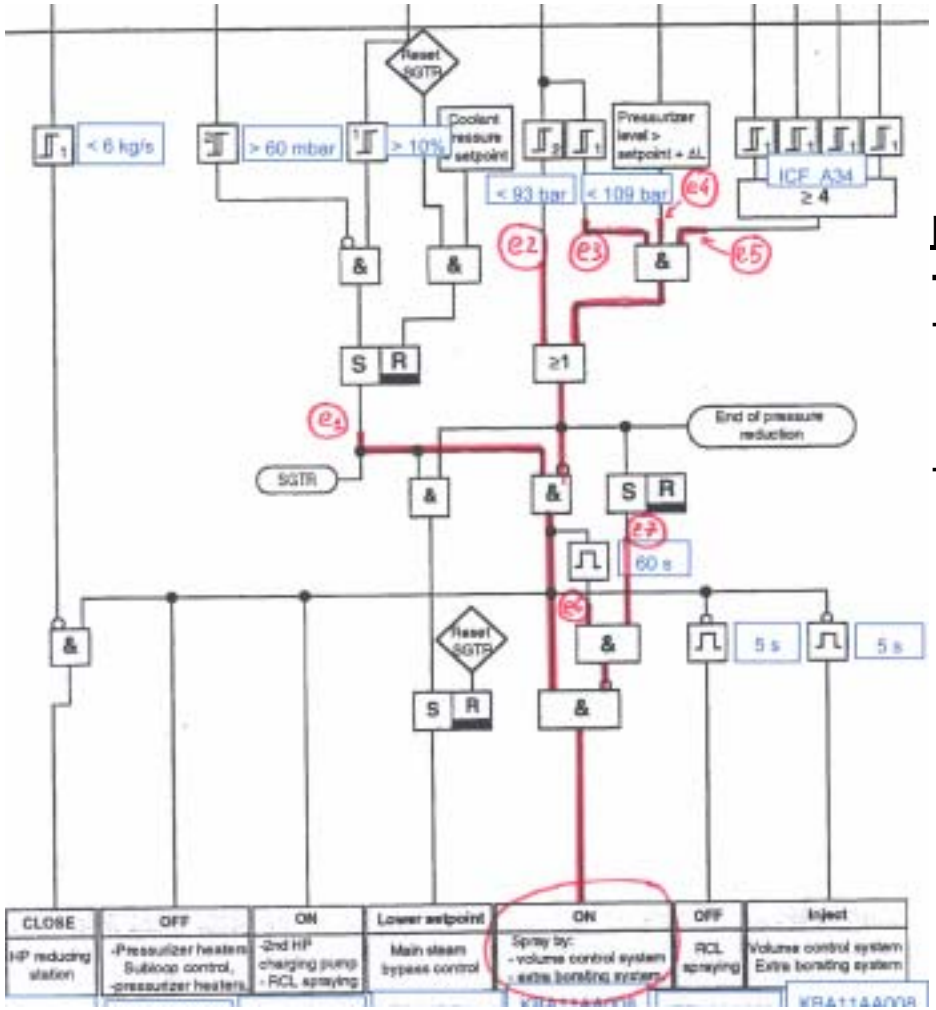
- en fonction des connections et des dépendances amonts certaines catégories peuvent, pour certaines instances de modules, ne pas être atteignables (il faudra pouvoir détecter formellement ces catégories inatteignables/vides)





Critères de couverture des tests fonctionnels

2nd critère: couverture des conditions logiques élémentaires



principe:

-sélectionner une sortie binaire

-remonter tant que l'on traverse des modules de logique combinatoire

-identifier les catégories : chaque signal binaire impliqué déclenche la sortie sélectionnée, les autres restant inchangés.

-> catégories inatteignables/vides …

Mise en œuvre dans le projet BE-SECBS

Projet européen

- Comparaison des méthodes d'évaluation des AS de plusieurs pays
- Basé sur un cas réaliste fourni par un industriel du secteur nucléaire

IRSN

- Mise en œuvre de la méthodologie d'évaluation classique
- Opportunité d'expérimenter sur un cas réaliste en taille et composition
  -> mesure objective de couverture des tests fonctionnels

Cas et documents disponibles

- Fonctions de limitation d'un réacteur nucléaire (cf schémas)
- Docs : exigences (schémas informels + texte libre)
  conceptions, codage ...
  scénarios des tests de validation fonctionnelle -> couverture ?

L'outil de génération de tests GATeL (1)

Entrée de l'outil : programme LUSTRE

- spécification fonctionnelle du code à tester
  - formalisation par l'évaluateur des exigences informelles (indépendance de point de vue)
  - modèle partiel : seuls les fonctions à évaluer doivent être détaillées
- contraintes appliquées par l'environnement sur les entrées
  - ex : gamme des entrées analogiques, vitesse max de variation, dépendance par rapport aux sorties passées...

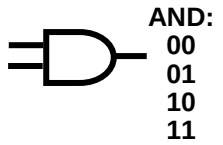
Intérêt de LUSTRE

- bien adapté à la description des aspects logiques, numériques et temporels des systèmes de sûreté
- sémantique bien définie -> exploitation non ambiguë

L'outil GATeL (2)

Expression des objectifs de test dans GATeL

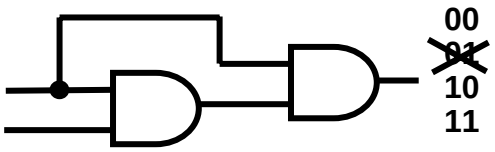
- « dépliage » interactif d'opérateurs en N cas ex:



- directives permettant de décrire des catégories moins locales
  - implémentation quasi directe du 2ème critère
- écriture d'une formule logico-temporelle quelconque en LUSTRE..

Résolution en deux étapes

- catégories atteignables



-1 test (scénario entrées/sorties) par catégorie atteignable

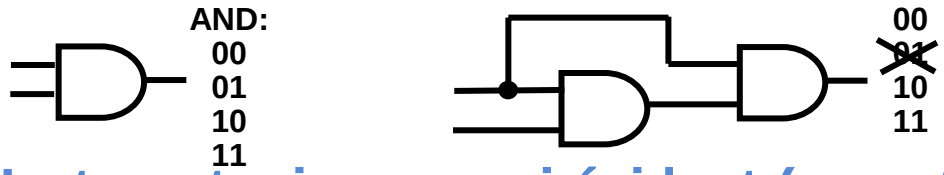
Contact GATeL : bruno.marre@cea.fr

Mesure de la couverture des tests fonctionnels (1)

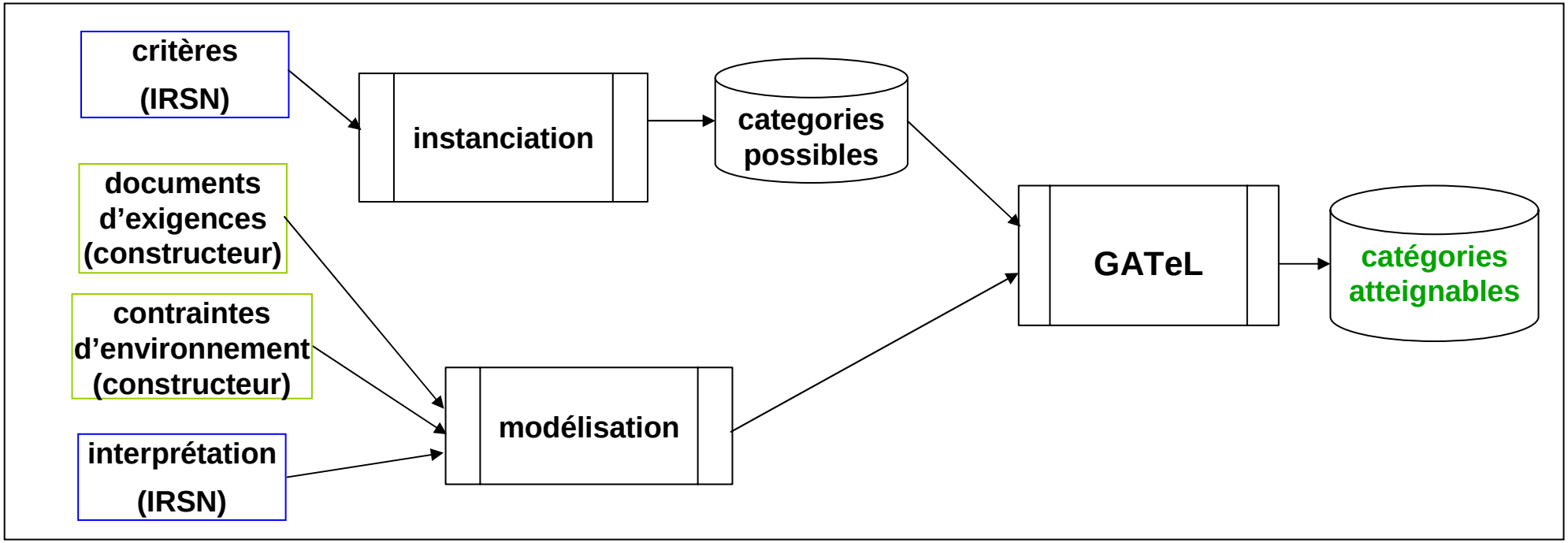
Etape 1

Déterminer les catégories atteignables pour chaque critère (local ou global) appliqué à chaque objet (composant, sortie booléenne).

Rappel :



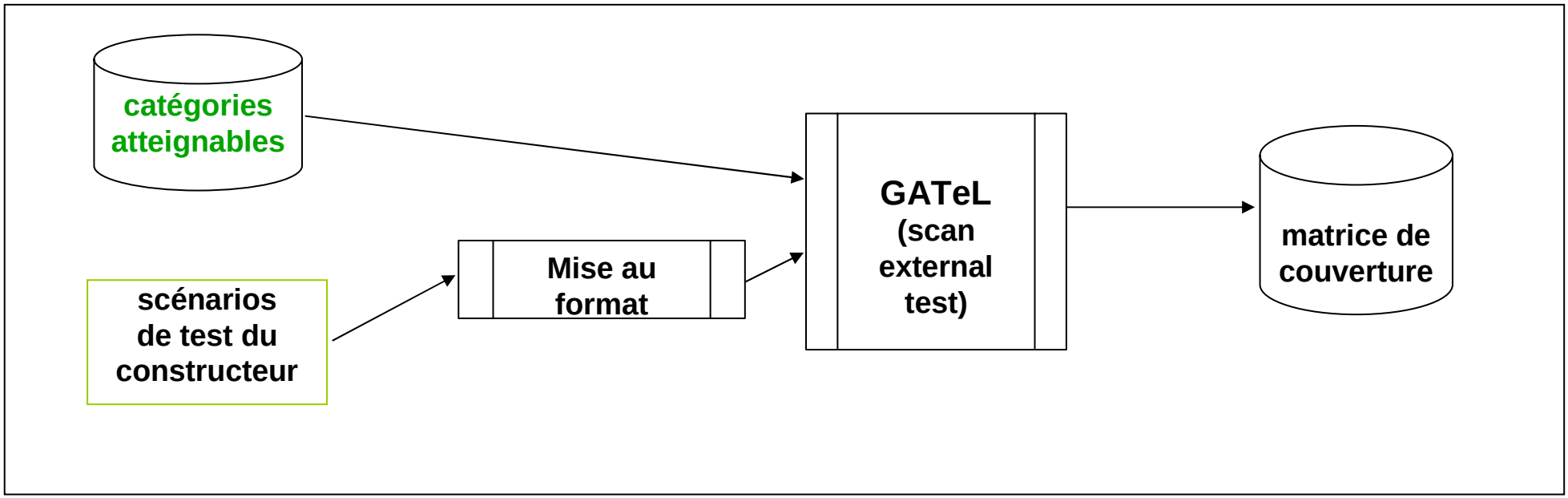
... mais ce n'est pas toujours aussi évident (aspects temporels ..)



Mesure de la couverture des tests fonctionnels (2)

Etape 2

Déterminer les éventuelles catégories atteignables non couvertes par les tests du constructeur

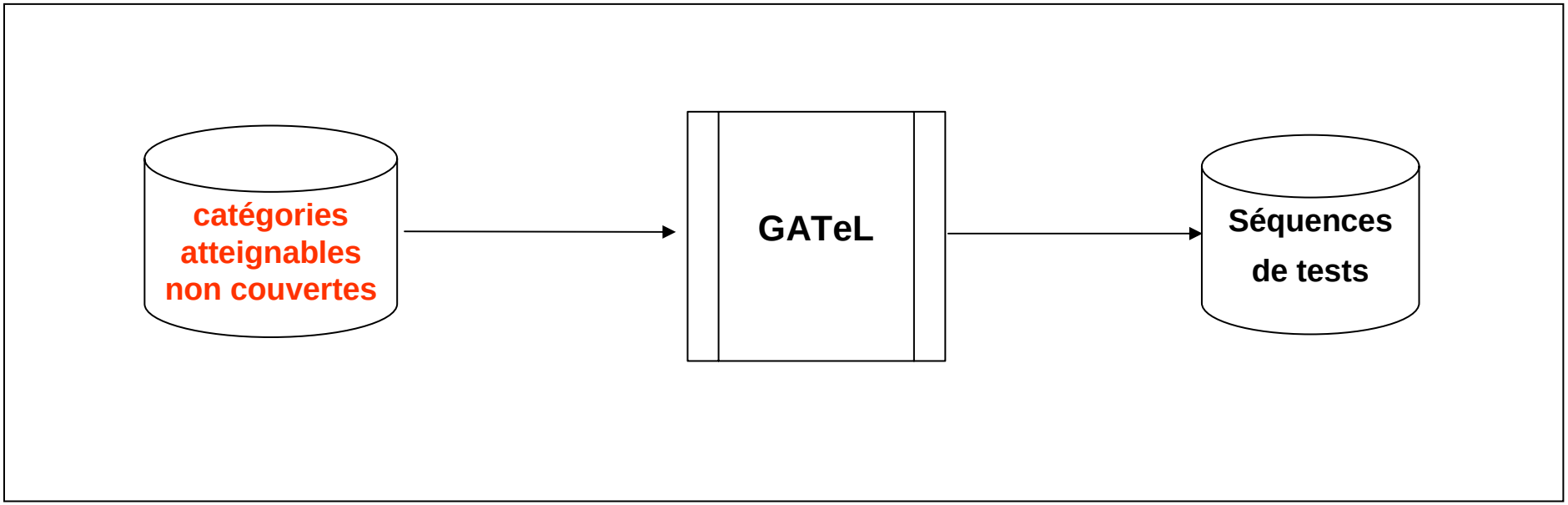


Remarque : avant la disponibilité de la fonction « scan », un autre outillage a été utilisé pour cette étape (exécution des jeux de test sur l'outil de simulation CLAIRE + filtrage des résultats).

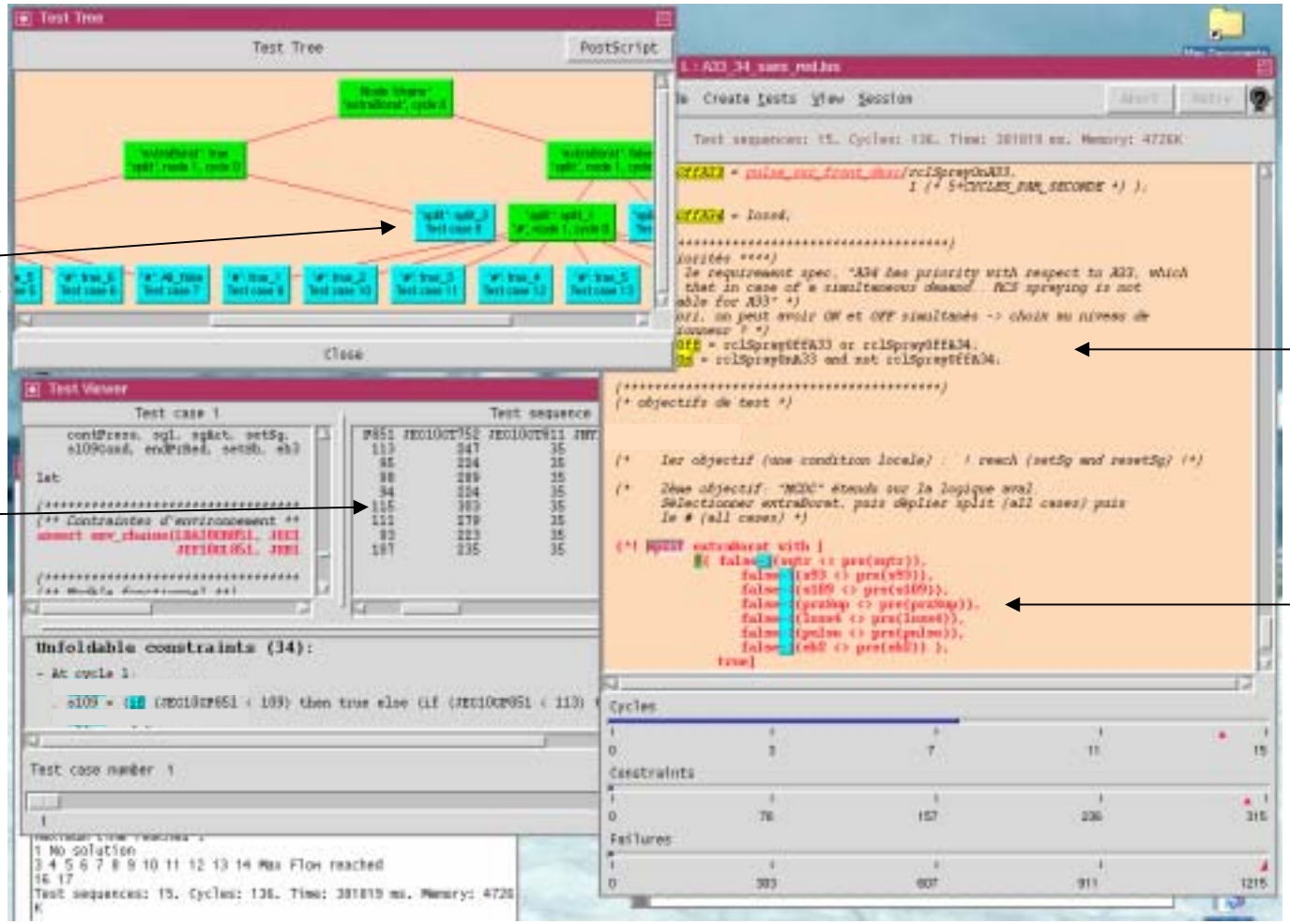
Mesure de la couverture des tests fonctionnels (3)

Etape 3

Pour chaque catégorie atteignable non couverte par les tests du constructeur, produire un scénario de test (met une lacune en évidence, sans hypothèse de bon fonctionnement de l'outillage).



GATeL



catégories atteignabes

scénario de test pour une des catégories atteignables

modèle fonctionnel

objectif de test

Intérêts de l'approche

Critères compréhensibles par les non-informaticiens

-> adaptée aux tests de validation fonctionnelle

Test indépendant de la chaîne de développement (diversification..) et de l'interprétation des exigences par le développeur

Ecriture naturelle du modèle à partir des exigences système

-> traduction aisée des diagrammes

-> niveau de détail modulable selon ce qu'on veut couvrir

Définition formelle des catégories et calcul automatique

-> mesure objective de la couverture

Calcul des scénarios manquants

Passe à l'échelle des logiciels critiques nucléaires

IRSN

Perspectives

Prise en compte des nombres en virgule flottante

- actuellement, GATeL offre des entiers de taille arbitrairement grande

-> pbs d'interprétation des résultats (remplacement réels par virgule fixe)

-> idem virgule flottante...

Test automatique intensif

Objectif : explorer chaque catégorie (implémentation..)

Méthode envisagée:

- GATeL génère N tests pour chaque catégorie atteignable
- CLAIRE les concrétise et les exécute
  - exécute sur station de travail le véritable code binaire, non instrumenté
  - simule le microprocesseur et l'environnement matériel du binaire
- GATeL permet d'implémenter l'oracle (barres d'erreur, glissement..)

-> quelle stratégie d'exploration des catégories ?