

VÉRIFICATION ET VALIDATION

Techniques de test

VV030
v151b

2014-09-03

Luc LAVOIE
Département d'informatique
Faculté des sciences



Luc.Lavoie@USherbrooke.ca
<http://info.usherbrooke.ca/llavoie>

TECHNIQUES DE TEST

QUE FAUT-IL POUR FAIRE UN TEST ?

- Le code exécutable du composant
- Un banc d'essai
- Un jeu d'essai
- Une procédure d'essai
- Un cahier d'essai
- Un guide d'interprétation des résultats
- ... et le plus souvent un *oracle* !

TECHNIQUES DE TEST

COMPOSITION D'UNE PROCÉDURE DE TEST

- Objectifs visés
 - exigences, critères, mesures
- Description et motivation de la stratégie retenue
- Sélection et motivation des techniques utilisées
- Jeux de test
- Description du banc d'essai
- Conditions
 - démarrage
 - arrêt
 - reprise
 - ...
- Démarche (liste des étapes et des tâches)
- Code (ou spécification détaillée)

TECHNIQUES DE TEST

APERÇU DES TECHNIQUES

- TFD – tests fonctionnels dyn.
 - Analyse axiomatique
 - Types abstraits algébriques
 - *Algèbre relationnelle (IFT 487)*
 - *Algèbre des processus (IGL 501)*
 - Analyse partitionnelle
 - Tests aux limites
 - Tests syntaxiques
 - Tests aléatoires
 - Graphes fonctionnels
 - Graphes de causes à effets
 - Analyse transactionnelle
 - Algorithmique qualitative
 - Analyse combinatoire
 - Survol
- *TFS – tests fonctionnels stat.*
 - *non couverts en IGL 601*
- TSD – tests structurels dyn.
 - Analyse de flot
 - Flot de données
 - Flot de contrôle
 - *Tests mutationnels*
 - *forts*
 - *faibles*
 - *Tests évolutionnistes*
 - *Analyse de sensibilité*
 - *Analyse des domaines finis*
 - *Exécution abstraite*
- *TSS – tests structurels stat.*
 - *non couverts en IGL 601*

TECHNIQUES DE TEST

NOTES ET RAPPELS

- Toute technique fonctionnelle peut être utilisée dans un contexte structurel
- Pas l'inverse
- Pourquoi ?

- Dans le cadre du cours, nous distinguerons
 - nature (fonctionnelle/structurelle)
 - instrumentation (externe/interne)

TECHNIQUES DE TEST

TESTS FONCTIONNELS DYNAMIQUES (TFD)

- Analyse axiomatique
 - Types abstraits
 - *Algèbre relationnelle*
 - *Algèbre des processus*
- Analyse partitionnelle
 - Tests aux limites
 - Tests aléatoires
 - Tests syntaxiques
 - Graphes de causes à effets
 - Graphes fonctionnels
 - Analyse transactionnelle
 - Algorithmique qualitative
- Analyse combinatoire

TECHNIQUES DE TEST (TFD.AP)

ANALYSE PARTITIONNELLE

Propriétés

- Générales
 - non vide
 - non intersection
 - complétude
- Spécifiques aux tests
 - Chaque classe doit être interprétable en fonction
 - de la spécification (tests fonctionnels)
 - de la conception (tests structurels)
 - Chaque représentant est générable efficacement

Limites

- Fondement adéquat de la partition
- Couverture statistique faible
- Existence, choix et qualité des substituts
- Existence, choix et qualité de l'oracle

TECHNIQUES DE TEST (TFD.AP.LIM)

TEST AUX LIMITES

- Technique
 - Pour chaque donnée, extraire les valeurs limites de son domaine de définition
 - Combiner l'ensemble de ces valeurs pour établir les données de test selon l'une des approches
 - minimale (somme)
 - maximale (produit)
 - représentative (sélection)
- Évaluation
 - Simple
 - Couverture rarement suffisante
 - peut souvent facilement être complétée par des tests aléatoires

TECHNIQUES DE TEST (TFD.AP.ALEA)

TESTS ALÉATOIRES

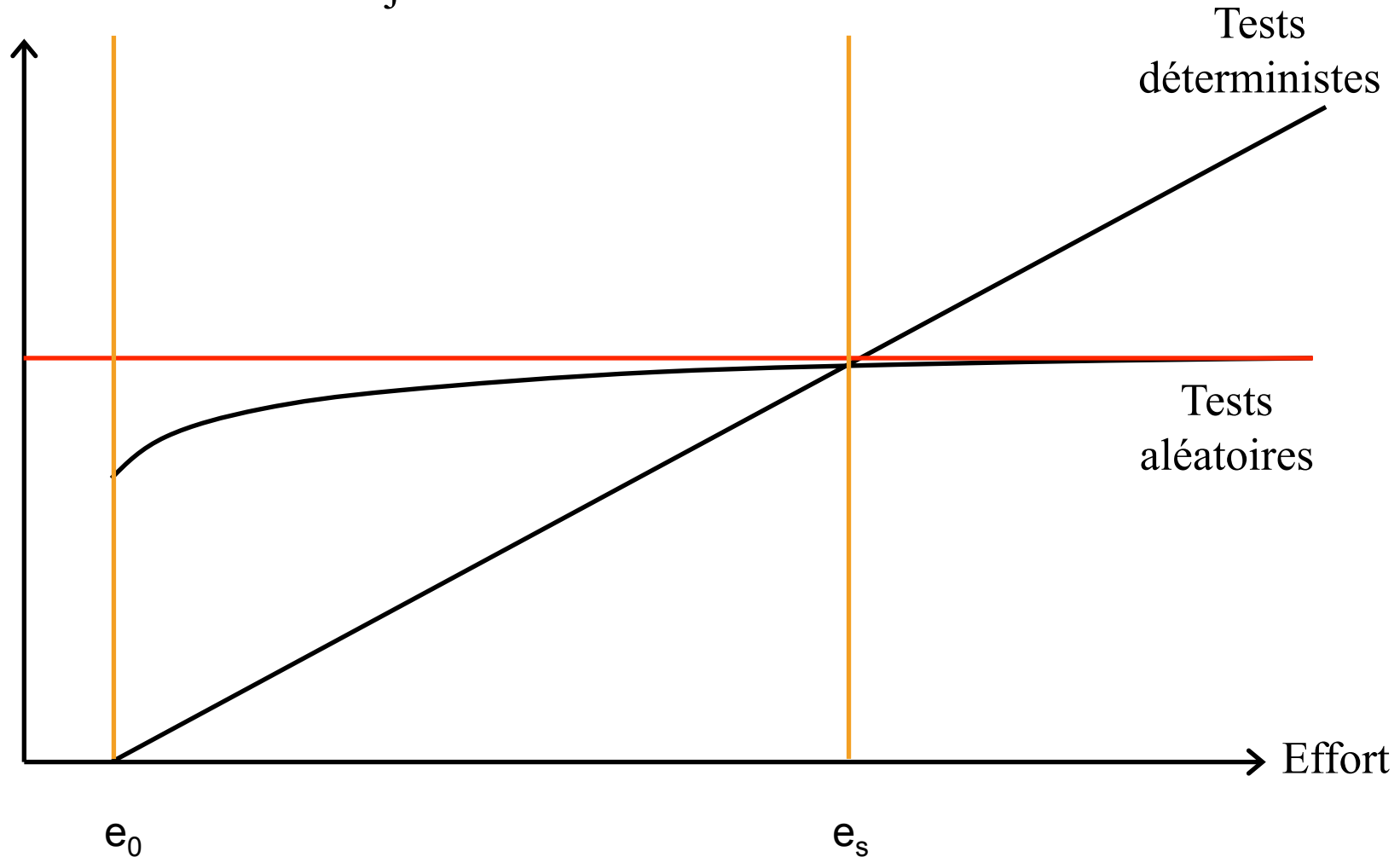
- Technique
 - Construire les DT à partir d'un échantillonnage uniforme des domaines de définition
- Évaluation
 - Facilement automatisable
 - Tests objectifs
 - Complémentaires des tests aux limites
 - En l'absence d'oracle déterministe, le niveau de confiance est le plus souvent difficile à quantifier
 - Il est parfois difficile de garantir l'indépendance des variables, donc la validité du test

TECHNIQUES DE TEST (TFD.AP)

RÉPARTITION DES TESTS – DÉTERMINISTES ET ALÉATOIRES

RENDEMENT DE L'EFFORT DE DÉVELOPPEMENT

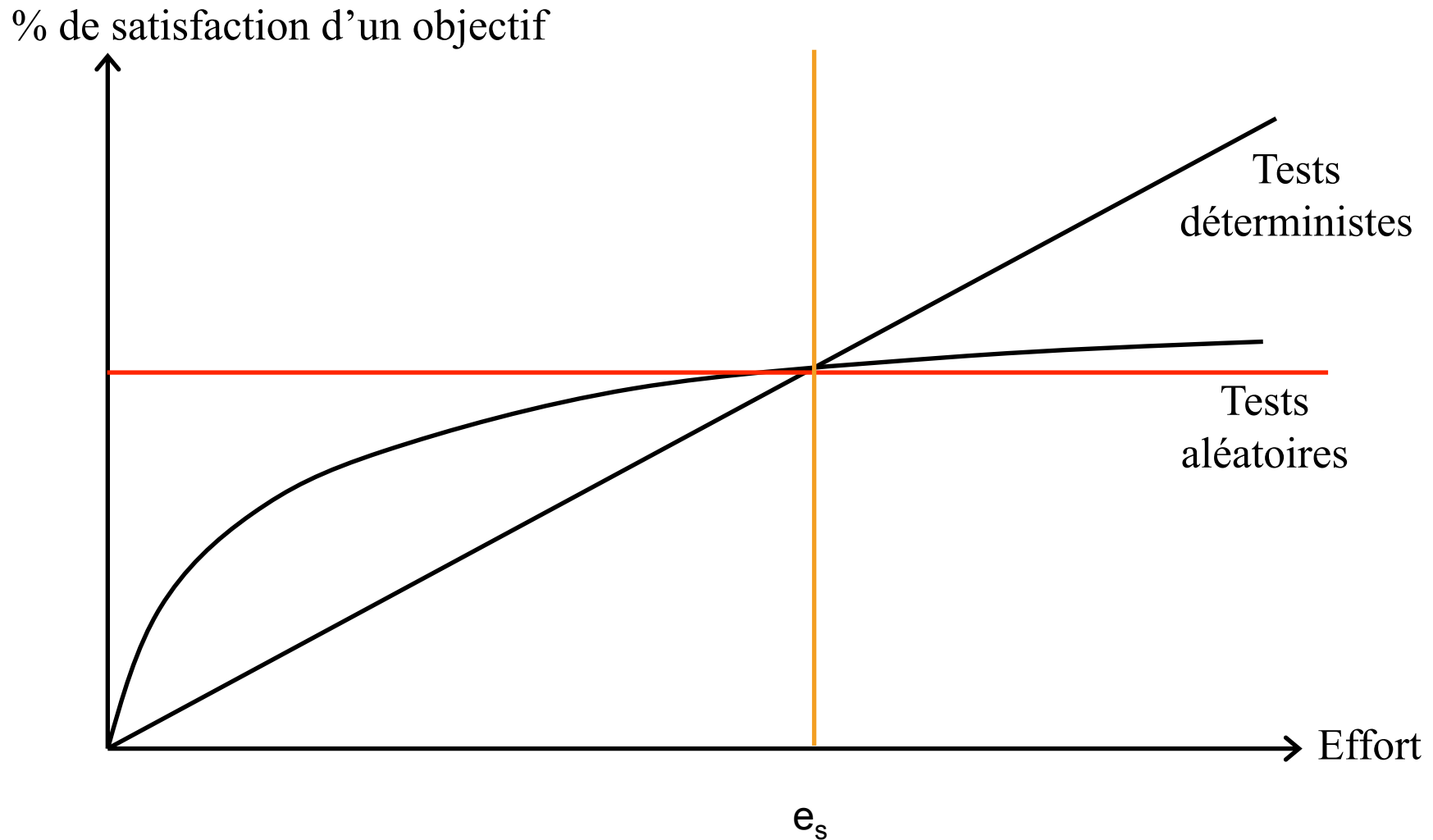
% de satisfaction d'un objectif



TECHNIQUES DE TEST (TFD.AP)

RÉPARTITION DES TESTS – DÉTERMINISTES ET ALÉATOIRES

RENDEMENT DE L'EFFORT DE CALCUL



TECHNIQUES DE TEST (TFD.AP.ALEA)

TESTS ALÉATOIRES ET TESTS AUX LIMITES

- Les tests aléatoires comme un cas particulier de tests aux limites
- Même un « mauvais » oracle, dans la mesure où ses caractéristiques sont connues et quantifiées, peut donner un bon niveau de confiance, si on y met le temps !

TECHNIQUES DE TEST (TFD.AP.SYN)

TESTS SYNTAXIQUES

○ Technique

- Décrire l'univers des problèmes par une grammaire formelle (régulière ou hors contexte)
- Produire des DT pour couvrir tous les noeuds terminaux de l'arbre de dérivation de la grammaire

○ Évaluation

- Limités aux données pouvant être décrites formellement
- Nombreux outils disponibles
- Également utile pour construire des oracles

TECHNIQUES DE TEST (TFD.AP.SYN)

TESTS SYNTAXIQUES – EXEMPLE 1

Jeu ::= Sentence Dimensions Coefficients Notes .

Sentence ::= "Succes" | "Echec" | "Arret" .

Dimensions ::= Jugues Figures .

Jugues ::= ENTIER .

Figures ::= ENTIER .

Coefficients ::= *[CoeffDiff]* .

CoeffDiff ::= FLOTTANT .

Notes ::= *[Note]* .

Note ::= FLOTTANT .

TECHNIQUES DE TEST (TFD.AP.SYN)

TESTS SYNTAXIQUES – EXEMPLE 2

Jeu ::=

Sentence Dimensions (^j,f)

Coefficients (\f^cd)

Notes (\j,f^n) .

Sentence ::=

"Succes" | "Echec" | "Arret" .

Dimensions (^j,f) ::=

Juges (^j)

Figures (^f) .

Juges (^j) ::=

ENTIER (^j) .

Figures (^f) ::=

ENTIER (^f) .

Coefficients (\f^cd) ::=

*[CoeffDiff (^c)

{cd(#f) := c}

]*f* .

CoeffDiff (^c) ::=

FLOTTANT (^c) .

Notes (\j,f^n) ::=

*[

*[Note (^x)

{n(#j,#f) := x}

]*f*

]*j* .

Note (^x) ::=

FLOTTANT (^x) .

TECHNIQUES DE TEST (TFD.AP.GFO)

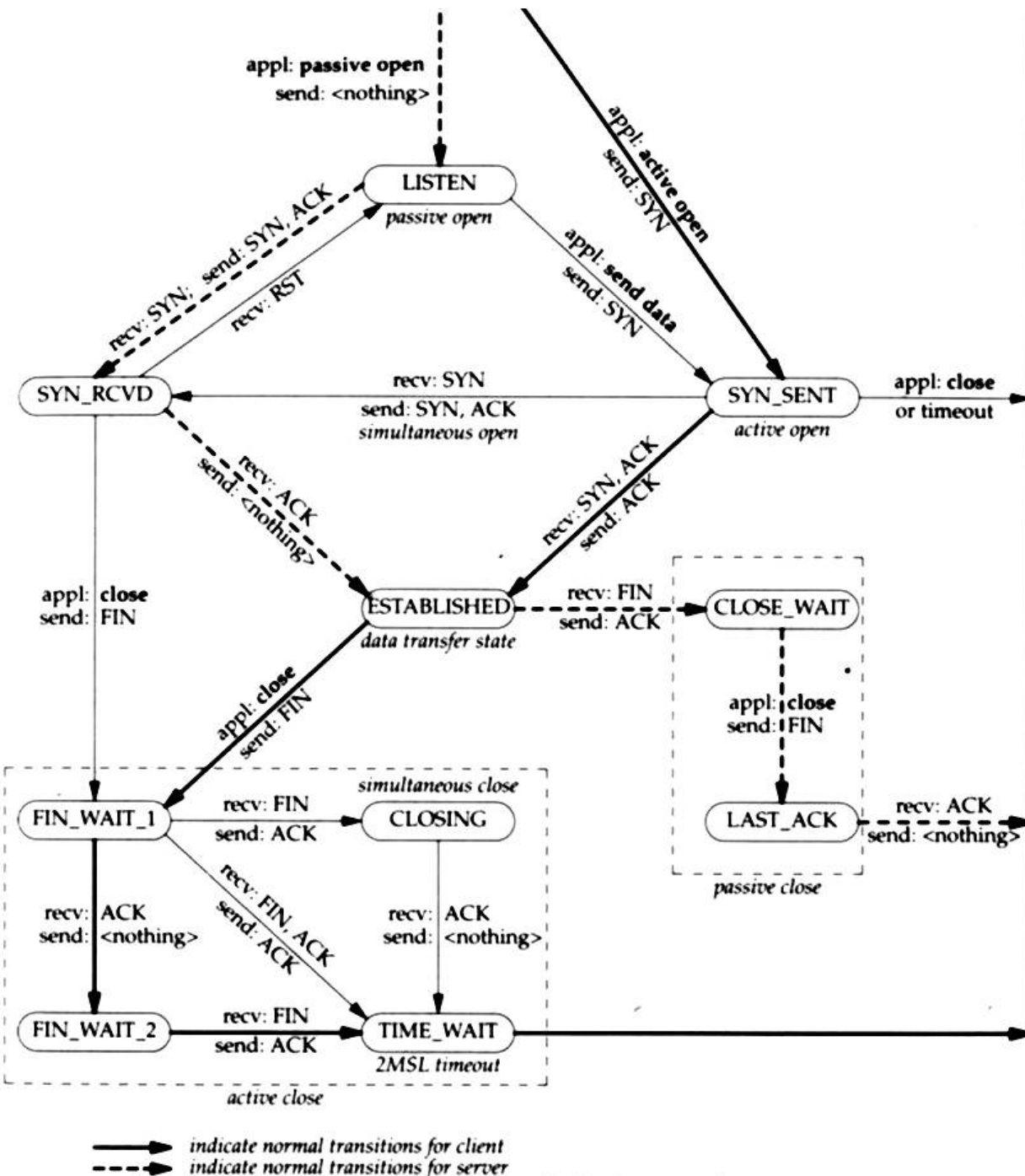
GRAPHES FONCTIONNELS

○ Technique

- Construire un graphe du comportement du logiciel (typiquement un automate).
- Les noeuds représentent un sous-ensemble des états du logiciel.
- Les DT couvrent en tout ou en partie les chemins de ce graphe.

○ Évaluation

- Particulièrement adaptée aux systèmes à haute interaction avec l'environnement :
 - diagrammes de séquence, d'interaction.
 - protocoles.



indicate normal transitions for client
 indicate normal transitions for server

GRAPHES FONCTIONNELS EXEMPLE

Diagramme d'états Le protocole TCP

- L'automate est utilisé
 - à l'entrée pour prédire l'état sur la base de la DT (en fonction de la spécification);
 - pour ausculter le composant et connaître l'état calculé (par le composant);
 - à la sortie pour déduire l'état résultant (en fonction de la spécification).
- Le test signale une erreur si les trois valeurs ne sont pas compatibles.
- Ceci suppose une fonction d'auscultation devant souvent être ajoutée à la spécification du composant — **d'où l'importance de concevoir et planifier les tests avant la conception du composant lui-même.**

GRAPHES FONCTIONNELS APPLICATION TYPIQUE

Particulièrement utile pour les tests d'intégration de bas niveau, centrés sur un composant critique.

Permet de vérifier le comportement d'un composant en contexte.

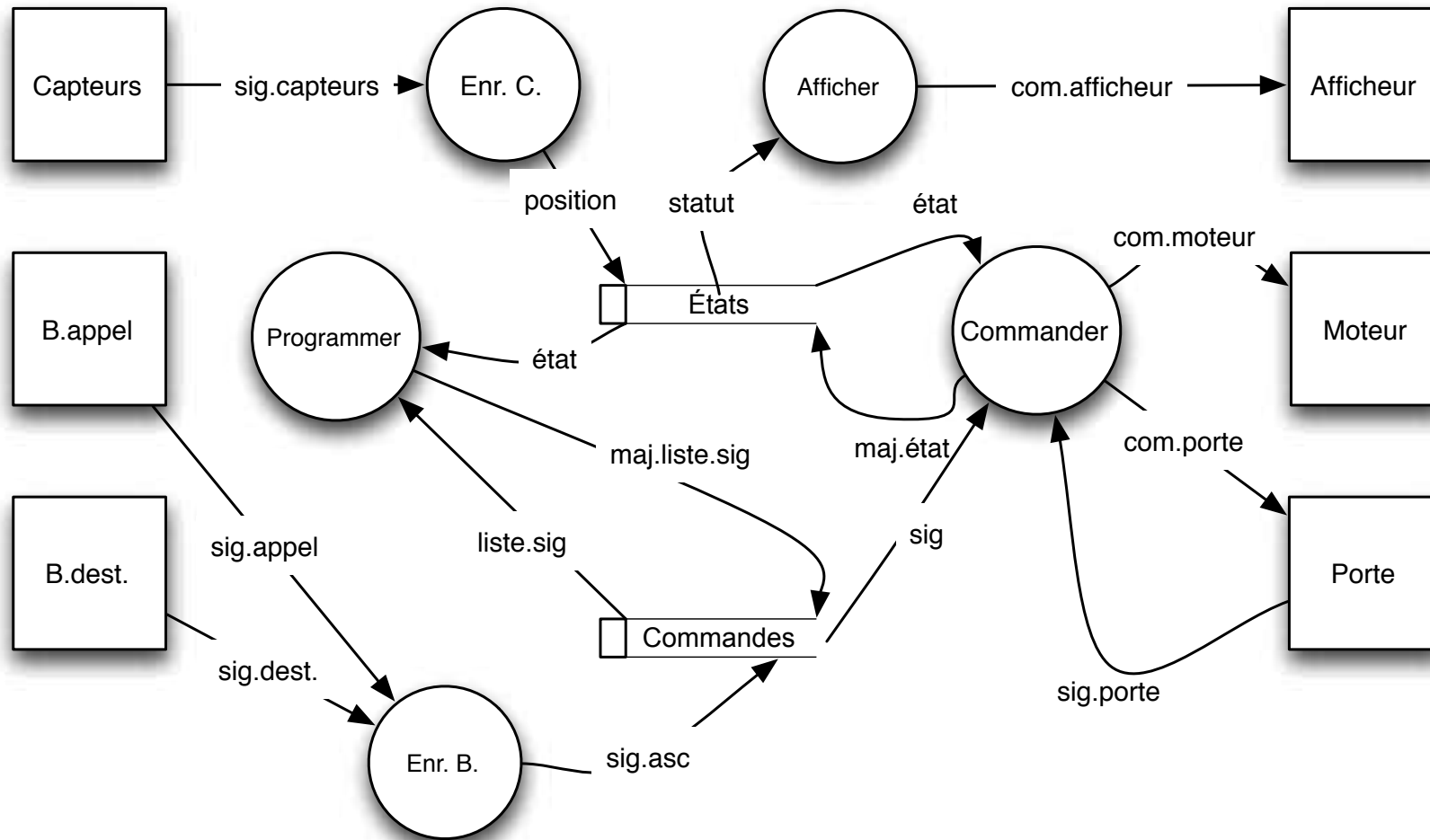
TECHNIQUES DE TEST (TFD.AP.TRANS)

ANALYSE TRANSACTIONNELLE

- Technique
 - Construire le DFD correspondant
 - L'unité de base est la transaction
 - Découpe de façon abstraite le flot
 - Générer des DT couvrant le DFD
- Évaluation
 - Technique fonctionnelle par excellence
 - Facilement composable
 - Permet d'obtenir facilement d'excellents taux de couverture
 - Les cas non couverts sont souvent inaccessibles par cette technique (erreurs, exceptions, etc.)

TECHNIQUES DE TEST (TFD.AP.TRANS)

ANALYSE TRANSACTIONNELLE - EXEMPLE



- Le DFD est utilisé pour choisir les flux devant faire l'objet d'une surveillance.
- On ajoute un processus artificiel sur chaque flux surveillé (ce processus reçoit les transactions et les retransmet de façon transparente).
- Chaque processus artificiel intrant transmet une commande de vérification associée à une transaction aux processus artificiels extrants.
- Chaque processus artificiel valide ses transactions intrantes sur la base des commandes de vérification reçues;
- Le test signale une erreur si les trois valeurs ne sont pas compatibles.
- Ceci suppose que chaque transaction d'origine (issue d'un agent) est dotée d'un marqueur unique et que chacune des transactions internes est annotée par la liste des marqueurs des transactions qui la déterminent — **d'où l'importance de concevoir et planifier les tests avant la conception du composant lui-même.**

ANALYSE TRANSACTIONNELLE APPLICATION TYPIQUE

Particulièrement utile pour les tests d'intégration de haut niveau, centrés sur le comportement des sous-systèmes entre eux.

Permet de vérifier le flux des données entre les sous-systèmes (ou les composants).

TECHNIQUES DE TEST (TFD.AP.GCE)

GRAPHES DE CAUSE-EFFET

- Technique
 - Représenter les relations causales entre les entrées et les sorties par un graphe
 - Déterminer les tests, les DT et les critères applicables
 - table de décision
 - arbre de décision
 - etc.
- Évaluation
 - Mise en pratique facile si la spécification est adéquate, laborieuse sinon
 - Supporte mal les changements aux spécifications

TECHNIQUES DE TEST (TFD.AP.GCE)

GRAPHES DE CAUSE-EFFET – EXEMPLE

| Chaufferie centralisée | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-------------------------------|------|-----|-----|-----|-----|-----|------|-----|---|
| Chauffage | on | | | | | | off | | |
| Eau chaude | on | | | off | | | on | off | |
| Thermostat réservoir | haut | | bas | | - | | haut | bas | - |
| Thermostat pièce | h | b | h | b | h | b | - | - | - |
| Chaudière on | | oui | oui | oui | | oui | | oui | |
| Pompe on | oui | oui | oui | oui | oui | oui | | oui | |
| Valve (3V) | B | B | A | C | B | B | A | A | - |

TECHNIQUES DE TEST (TFD.AP.QUAL)

ALGORITHMIQUE QUALITATIVE

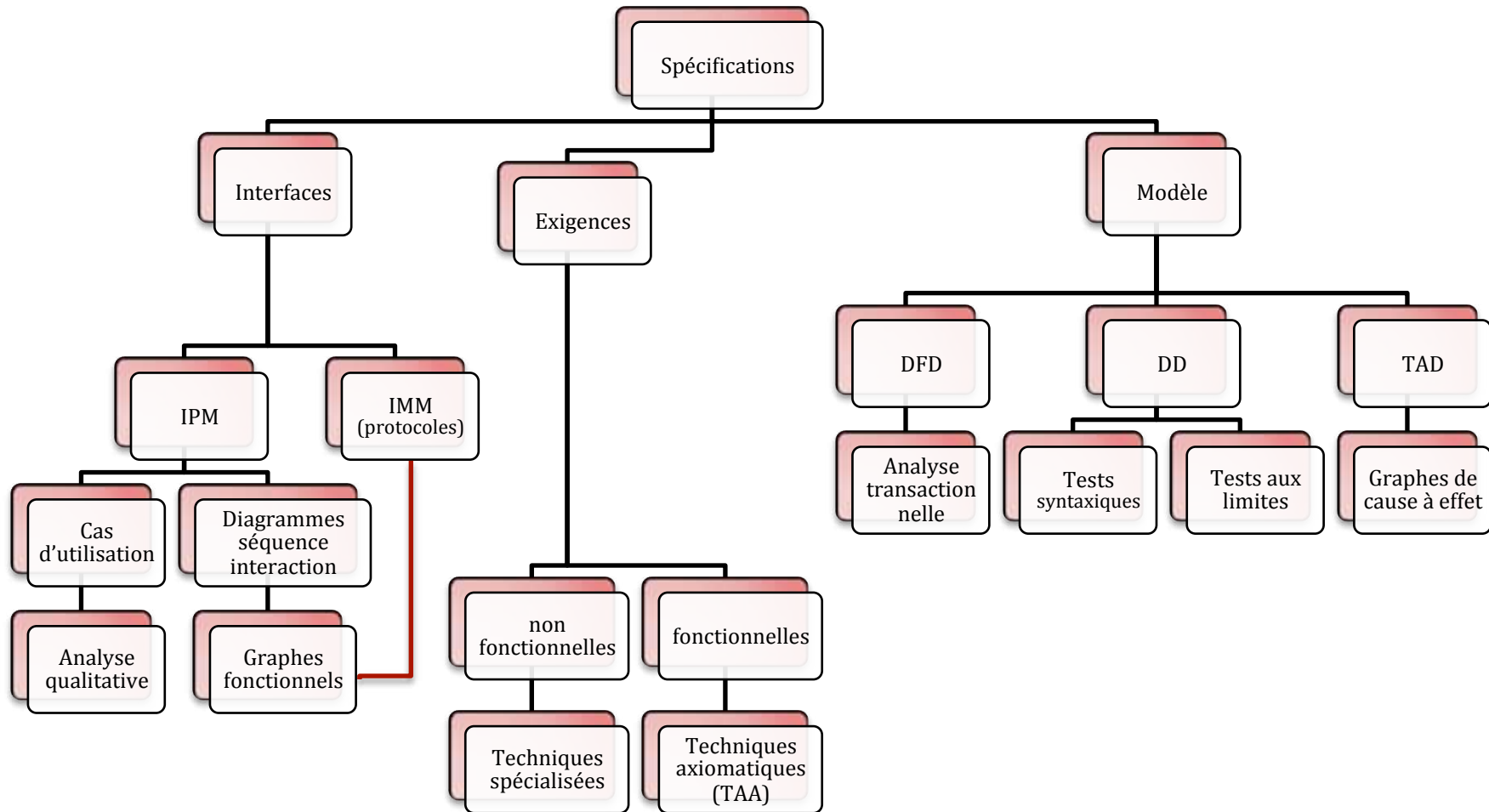
- Technique
 - Construire un graphe dont les sommets représentent des classes d'exécution
 - cas d'utilisation (CU)
 - modèles mathématiques (MM)
 - etc.
 - Générer des DT permettant de couvrir le graphe
 - CU : histoires de cas
 - MM : exemples de problèmes
 - etc.
- Évaluation
 - Adaptée aux logiciels
 - de soutien interactif à une tâche
 - à fortes composantes numériques
 - faisant appel à des algorithmes heuristiques
 - Ressemble à la technique de couverture de graphe fonctionnel
 - Permet d'avoir une vue qualitative sur la dynamique du logiciel

AUTRES TFD

- Analyse axiomatique
 - voir module séparé
- Analyse combinatoire
 - voir module séparé

TECHNIQUES DE TEST

TFD - SYNTHÈSE



TECHNIQUES DE TEST

TESTS STRUCTURELS DYNAMIQUES (TSD)

- Propriétés communes des TSD
 - Réalisés à partir du code source
 - Instrumentés par modification du code
- Conséquences
 - L'objet du test est modifié (et sera vraisemblablement différent de l'objet mis en production).
 - La nécessité et la portée des tests de système sont accrues.

TECHNIQUES DE TEST

TESTS STRUCTURELS DYNAMIQUES (TSD)

- Analyse dynamique de flot
 - Flot de données
 - Flot de contrôle
- Autres (non couverts en IGL601)
 - Tests mutationnels
 - Tests mutationnels faibles
 - Analyse de sensibilité
 - Exécution abstraite
 - Analyse des domaines finis
 - Test évolutionniste

TECHNIQUES DE TEST (TSD.FLOT)

ANALYSE DYNAMIQUE DU FLOT

CRÉATION DU GRAPHE

- Chaque « unité algorithmique » (procédure, fonction, routine, méthode, etc.) est représentée par un graphe dont les sommets sont des « blocs de base »
- Un bloc de base est une suite d'instruction sans rupture de contrôle
 - chaque instruction est la seule continuation de la précédente
 - toutes les instructions partagent le même contexte
- La programmation structurée simplifie le processus

TECHNIQUES DE TEST (TSD.FLOT)

ANALYSE DYNAMIQUE DU FLOT

UTILISATION ET CLASSIFICATION DES CHEMINS

- Les chemins du graphe sont classés en deux catégories
 - exécutable
 - non exécutable
- Les chemins non exécutable sont exclus
 - souvent dus à une mauvaise programmation
- Les chemins exécutable servent de base à la génération de DT en fonction du
 - flot de contrôle
 - flot des données
- Les chemins exécutable sont « décorés » d'assertions propres aux flots considérés

TECHNIQUES DE TEST (TSD.FLOT.D)

ANALYSE DYNAMIQUE DU FLOT

COUVERTURE DU GRAPHE DE CONTRÔLE

- En fonction du flot des données
 - Tous les p-utilisateurs
 - Tous les p-utilisateurs /
quelques c-utilisateurs
 - Tous les utilisateurs
 - Tous les chemins
 - Toutes les définitions
 - Tous les c-utilisateurs /
quelques p-utilisateurs

TECHNIQUES DE TEST (TSD.FLOT.C)

ANALYSE DYNAMIQUE DU FLOT

COUVERTURE DU GRAPHE DE CONTRÔLE

- En fonction du flot de contrôle
 - Tous les nœuds — TER1
 - Tous les arcs — TER2
 - Portion linéaire du code suivie d'un saut (LCSAJ) — TER3,4,5...
 - Tous les chemins indépendants
 - Chemins limites et intérieurs
 - Chemins d'ordre 1
 - Critère MC/DC

TECHNIQUES DE TEST (TSD.FLOT)

ANALYSE DYNAMIQUE DU FLOT

ÉVALUATION

- Technique incontournable
 - base des techniques structurelles
- Impose une instrumentation importante au code source
- DT basés sur la source
 - pertinence garantie
 - peut réduire l'envergure effective des DT
- Il demeure pertinent de vérifier les résultats en regard des spécifications

Ordonner les obligations de test afin

- * de ne pas être obligé de les tester toutes à chaque fois ;
- * en ordre de dépendance de faisabilité
(si le test n'a pas de sens ou n'est pas réalisable en cas d'échec du test A, on fait le test A en premier) ;
- * en ordre croissant de coût.

Ordre proposé (dans le cas général) :

- * OT structurelles statiques
 - fc : ?
 - fd : def-use, etc.
- * OT fonctionnelles (exigences)
 - TAA avec DT respectant les antécédents
- * OT structurelles dynamique
 - fc : bloc, branchement, loop, condition, mcdc, ?
 - fd : ?
- * OT fonctionnelles (hors exigences)
 - TAA avec DT ne respectant pas les antécédents
- * OT non fonctionnelles