Outils de Modélisation 1 (parties 1& 2)

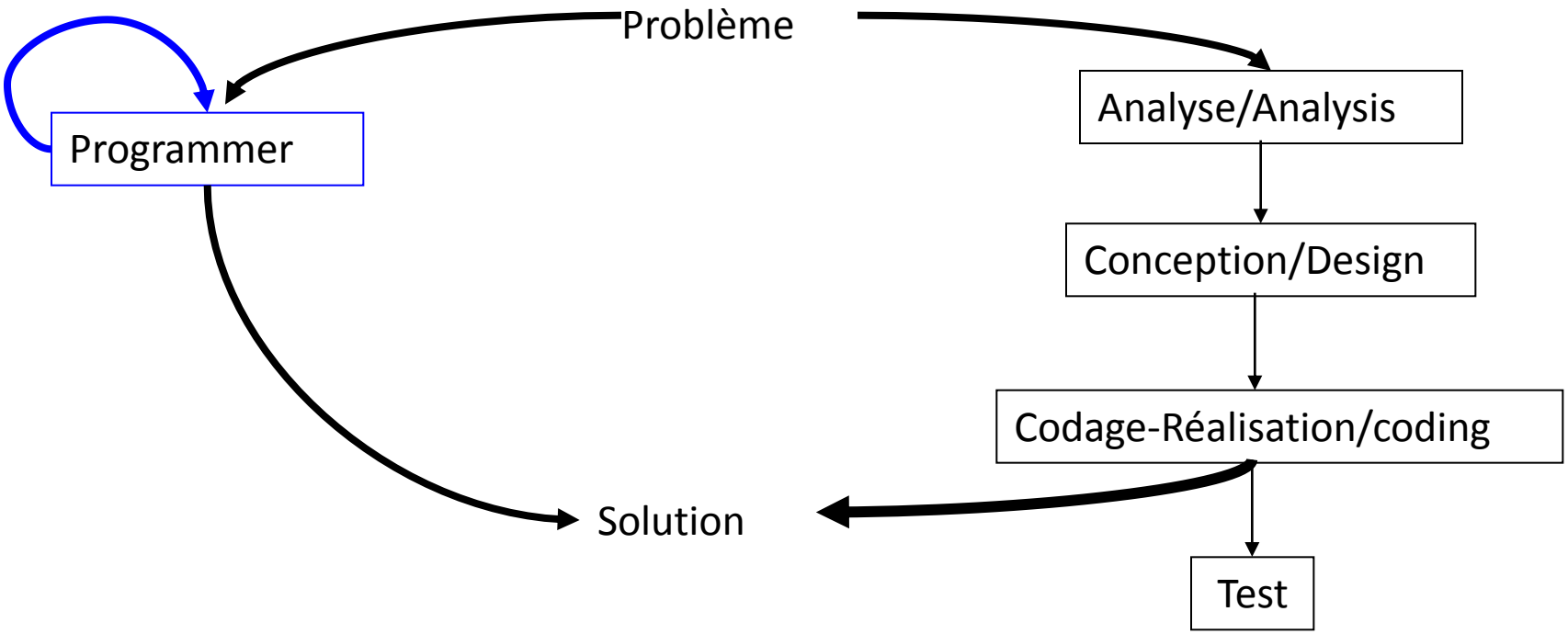
Cours de Master: M1 IA

Plan

- **Partie 1: Introduction (Systèmes critique, bogues célèbres, Langages formels, Spécification, Vérification);**
- **Partie 2: Modélisation du système (Aspects à modéliser dans un système, modèles pour la dynamique, systèmes à événement discrets, formalismes des systèmes à événement discrets)**
- Partie 3: Systèmes états-transitions
- Partie 4: Automates des systèmes infinis
- Partie 5: Logique Temporelle

But d’un développeur

- Trouver des solutions à des problème
- Développer des systèmes



Exemples de Systèmes

1. Systèmes répartis: réseaux, télécommunications . . .

2. Systèmes embarqués: téléphones mobiles, avions, fusées, automobiles

3. Systèmes domestiques: un magnétoscope, une machine à laver . . .

Systèmes critiques

- de plus en plus performants, de plus en plus miniaturisés et donc **de plus en plus complexes**

- systèmes critiques →**fiabilité indispensable**

Quelques bogues célèbres

Télécom: 1990

- un **patch non vérifié** dans le système d'exploitation
- une erreur dans un **switch** (en C)
- le réseau téléphonique de la côte Est des États-Unis a **été bloqué pendant 9h** !

Energie: 2003

• Panne d'électricité aux USA & Canada, General Electric. Cause. A nouveau : mauvaise gestion d'accès concurrents aux ressources dans un programme de surveillance.

Quelques bogues célèbres

Aéronautique:

- 1962: Perte d'itinéraire de la sonde Mariner 1 (NASA) au lancement.
  Cause. Erreur de transcription de copie papier vers code Fortran.

- 1996: Auto-destruction d'Ariane 5 (1er vol), 37 secondes après décollage. Cause: Conversion flottant 64 bits trop grand, vers entier 16 bits.

- 2004: Blocage du robot Mars River. Cause: Trop de fichiers ouverts en mémoire flash.

Medecine:

- 85–87: 5 morts par irradiations massives dues a la machine Therac-25.
  Cause: Conflit d'accès aux ressources entre 2 parties logicielles.

Quelques bogues célèbres

Informatique

06–08: Clés générées par OpenSSL et données cryptées non sures, impactant les applications l'utilisant (comme ssh). (http://linuxfr.org/news/d%C3%A9couverte-dune-faille-de-s%C3%A9curit%C3%A9-critique-dans-openssl-de-deb)
Cause. Générateur de nombres aléatoires d'OpenSSL casse.

1994: Bug du Pentium FDIV sur opérations en nombres flottants.
Cause. Algorithme de division erroné (découvert par Th. Nicely). 470 millions de $

78–95: Faille dans le protocole d'authentification de Needham-Schroeder.
Cause. Attaque « man in the middle » détectée par G. Lowe.

Bogues Quotidiens

- Ordinateur **perd un fichier**,
- **l'installation d'un nouveau logiciel** en rend un autre inutilisable
- certaines **options d'impression** refusent de fonctionner

Pourquoi ces bogues?

- La rédaction du **cahier des charges**:

  langage naturel peut être a source d'erreurs :

  1) les descriptions écrites peuvent être **ambigües**

  2) **mal interprétées** par les développeurs chargés de mettre en œuvre des solutions.

- La conception : une source potentielle d'erreurs:

  1) la complexité croissante des systèmes,
  2) Interactions avec d'autres systèmes,
  3) ou avec des parties du système

Solutions Possibles:

- Description claire du système = Méthode Formelle
- Spécification & Vérification du système

Spécification

- Description :

  1) **Abstraite** : pas trop de détail

  2) **Clarté et précision**: sans Ambiguïté

→ Langages (formalismes) de spécification ou de modélisation

Si on utilise des représentations graphique, on parle aussi de modélisation et de modèle

Langages de Sépcification

- Informels: langage naturel, table, …;
- Semi-formels: UML, DFD, DEA, …;
- Formels;

Langages Formels

Qualités d'un langage formel →

1. *L'expressivité*
2. *La vérifiabilité*
3. *L'abstraction*
4. *La lisibilité*

Aspects d'un langage formel:

1) Une syntaxe bien définie (alphabet, mots, règles d'écriture) et une sémantique formelle (interprétation)

2) Un fondement mathématique : permettre la vérification et la preuve

Langages Formels: classes

- Les approches basées sur **la logique** : théorie des ensembles (le langage Z, Vienna Development Method (VDM), méthode B), logiques temporelles (TRIO, TLA, TLCO), logiques d'ordre supérieur (Coq, PVS), logique linéaire.

- Les approches **algébriques** : algèbres de processus (CCS, LCS, LOTOS, RTL).

- Les approches par **modèles à états**: tels que les systèmes de transitions (avec StateCharts, SDL) et les Réseaux de Petri.

Vérification

- **Vérification: prouver qu'un système vérifie bien certaines spécifications (des propriétés).**

- **preuve mathématique du bon fonctionnement d'un modèle du système.**

Que peut on vérifier?

- On ne peut pas vérifier tout dans un système

- Plutôt il y'a un ensemble **limité** de propriétés qui sont possibles à vérifier et qui sont aussi nécessaires à vérifier.

Propriétés à vérifier?

- Atteignabilité: un certain état est atteignable durant l'exécution du système.

  Exemple: *Il existe un moyen pour que la machine serve du café*

- Vivacité: possibilité d'exécuter une action dans le système

  Exemple 1: *s'il reste du café et que l'utilisateur a mis le montant adéquat, la machine lui sert un café,*

  Exemple *2: le message envoyé d'un côté sera finalement délivré de l'autre côté d'un réseau*

Propriétés à vérifier?

Sûreté: le mauvais n'arrivera jamais.

- ***Exemple 1****: si le montant adéquat n'a pas été mis, la machine ne servira jamais de café,*
- ***Exemple 2****: un état de violation de l'exclusion mutuelle ne doit pas se présenter,*
- ***Exemple 3****:* la barrière doit obligatoirement se fermer un certain temps avant le passage du train.

- Équité, absence de blocage, …

Comment Vérifier: techniques?

- **Preuve de programme** : preuve de théorème, axiomes, …

- Model-checking sur un modèle de système;

- **Preuve d'équivalence** : Bisimulation entre modèles (exemple: modèle de spécification vs modèle de l'implémentation)

Limites :

Preuve = très couteuse, non automatisable

Model-checking= limités dans ses applications

test

- C’est quoi la spécification?
- C’est quoi un langage formel?
- Quels sont les types de langages formel?
- C’est quoi la vérification?
- Quelles sont les techniques de vérification?
- Que peut on vérifier ?

Partie (2)

Modélisation des systèmes

- Aspects à modéliser dans un système,
- Modèles pour la dynamique,
- Systèmes à événement discrets,
- Formalismes pour les systèmes à événement discrets

Aspects à modéliser (à spécifier) dans un système

- Structure: aspect statique en général dans le temps comme les structures de données, le code, la diposition physique du système

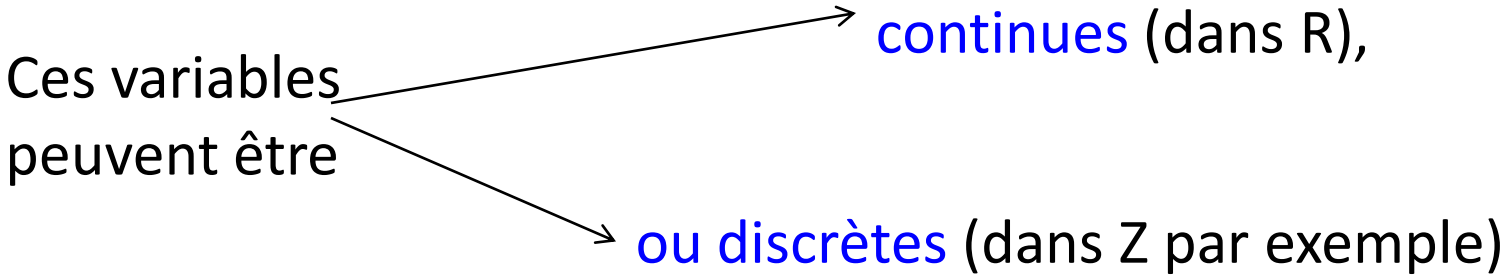
- **Dynamique**: aspect qui change dans le temps: l'état du système qui change dû à l'exécution d'actions ou l'arrivée des événements

- Formalismes adapté à la dynamique : algèbres de processus, systèmes de transition (Automates, réseaux de Petri) ,

- Formalismes adaptés à l'aspects statiques : modèles (Z, B) (spécifications algébriques, logiques)

Dynamique de systèmes: modélisation (1)

La dynamique d'un système est modélisé par un ensemble de variables d'états, qui changent durant le temps



La dynamique peut être : continue, échantillonnée, discrète, où à événements discrets

Dynamique de systèmes: continu (2)

1) Système continu: étudié souvent avec des modèles mathématiques, où on a:

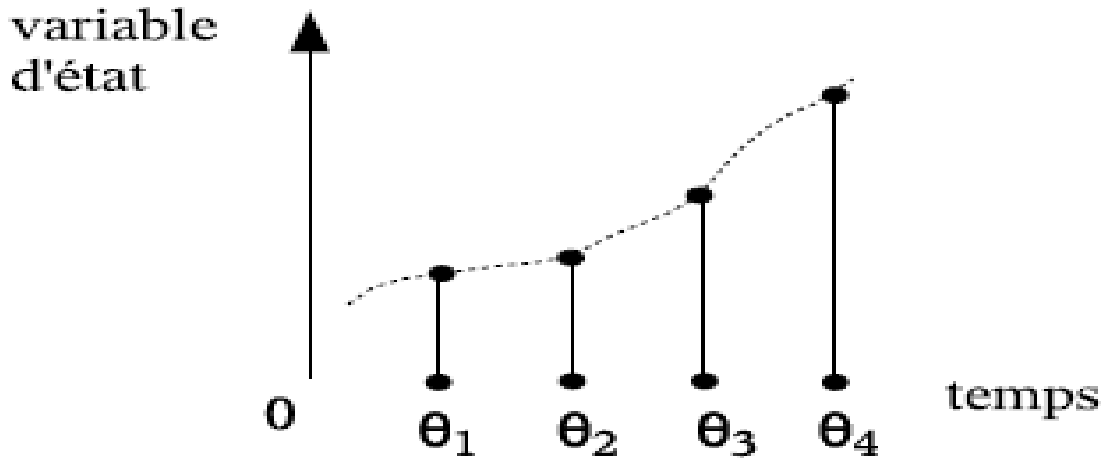
- Variables d'états sont continues et
- Le temps et aussi continu

Modèle mathématique= équation **différentielle** par exemple

Dynamique de systèmes: échantionné (3)

2) Système échantillonné:

- **Variables d'états:** sont continues et
- **Le temps:** est discrétisé (le temps est représenté sous forme d'une suite d'instants séparés l'un de l'autre)

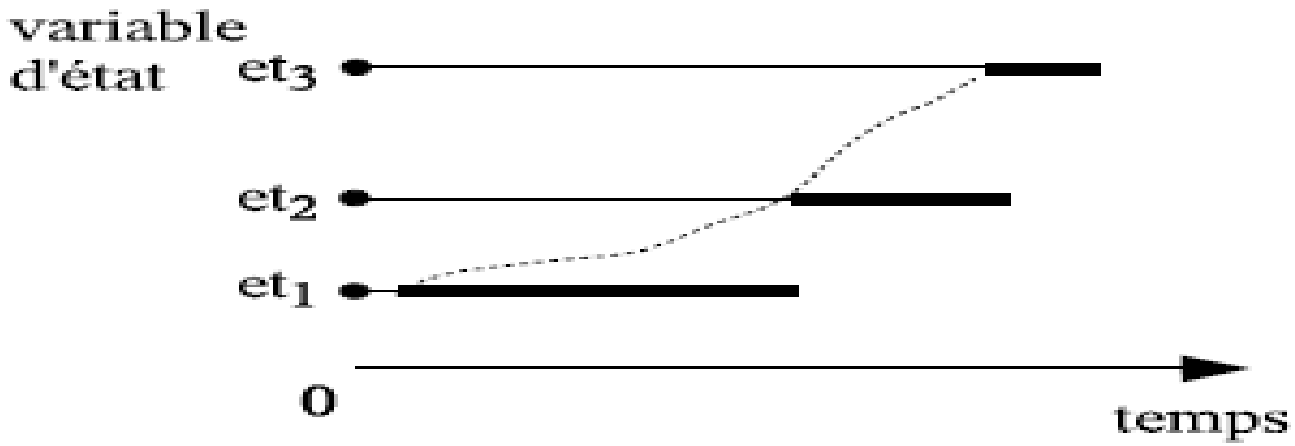


3) Système discret:

- Variables d'états sont discrètes
- et le temps est continu

Exemple1: lampe (**allumée**, **éteinte**), dispositif (**fonctionnel**, **défaillant**)

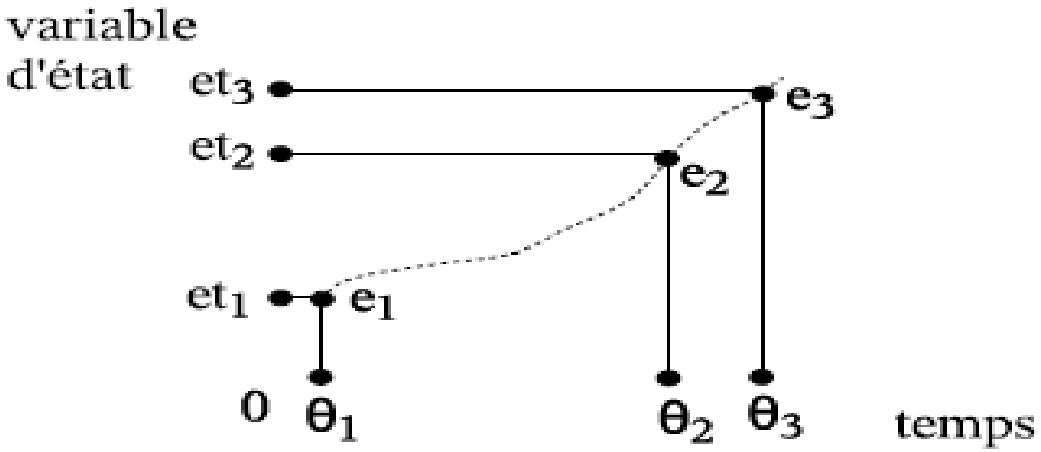
Exemple2: **Ascenseur** (rez-chaussée, 1er étage,…)



Dynamique de systèmes (5)

4) Système à événements discrets

- La dynamique du système est gérée par un ensemble d'événements discrets;
- Ces événements font transiter le système d'un état vers un autre.
- Donc le temps aussi est discrétisé car on s'intéresse aux instants aux les événements arrivent.



Les systèmes les plus étudiés

Remarque: En informatique on porte plus d'intérêt aux systèmes à **événements discrets**

Étudier les propriétés des systèmes à événements discrets=

1) **Énumérer tous les états,**

2) **et étudier les variables dans ces états.**

→ **Modèles formels pour ces systèmes ???**

Modèles pour les systèmes à événements discrets

- Des **systèmes états-transitions** : les automates (finis, hybrides, temporisés, …), les réseaux de Petri, …

- La **logique** : logique temporelle (CTL*, CTL, LTL, TCTL)

- les **algèbres de processus (CCS, CSP, …)**

Comment choisir un modèle?

Compromis: expressivité/facilité d'analyse

Expressivité :

- Représenter de nombreux systèmes
- Les représenter succinctement

Facilité d'analyse :

- Méthodes éfficaces...
- ... si elles existent

Test 2

- Citer certains aspects qu'on s'intéresse à modéliser dans un système?
- C'est quoi la dynamique (le comportement) d'un système? Donner des exemples
- Citer les modélisations possibles d'une dynamique d'un système. Pourquoi cette variété de modélisations?
- Citer quelques outils formels pour la modélisation de la dynamique d'un système
- Sur quelle base on favorise un outil par rapport un autre?