ALASCA — Architecture logicielles avancées pour les systèmes cyber-physiques autonomiques

© Jacques Malenfant

Master informatique, spécialité STL – UFR 919 Ingénierie

Sorbonne Université
Jacques.Malenfant@lip6.fr

Cours 3

Modélisation du comportement des systèmes de contrôle cyber-physiques

Objectifs pédagogiques du cours 3

- Adopter une vision **théorie des systèmes** pour analyser les systèmes cyber-physiques autonomiques et en comprendre les apports dans leur conception et leur mise en œuvre.
- Comprendre la modélisation **quantitative** des systèmes traditionnellement utilisée par les spécialistes de ce domaine et les automaticiens ainsi que les relations qui existent avec les systèmes cyber-physiques.
- Comprendre les différents modèles de systèmes : **continus**, **discrets**, **hybrides**, leurs interrelations et leur utilisation pour **spécifier** et **modéliser** les systèmes auto-adaptables.
- Approfondir plus spécifiquement les **systèmes hybrides** comme modèles de comportement des systèmes cyber-physiques autonomiques, et leurs principales déclinaisons développées dans le domaine de l'informatique théorique sous le vocable **automates hybrides**.

"All models are wrong – but some models are useful." *
— G.E.P. Box

G.E.P. Box, Dans *Robustness in the Strategy of Scientific Model Building*, R.L. Launer et G.N. Wilkinson, éditeurs, Academic Press, 1979.

* *« Tous les modèles sont faux, mais certains modèles sont utiles. »*

Plan

1. Théorie des systèmes
2. Systèmes hybrides
3. Modélisation hybride d'un système auto-adaptable
4. Panorama des modèles hybrides
5. Première décomposition du modèle de l'exemple
6. Modélisation des systèmes auto-adaptables

Auto-adaptabilité et théorie des systèmes I

- Pendant longtemps, l'informatique s'est intéressée à l'auto-adaptabilité mais à l'intérieur de ses frontières « cyber », c'est-à-dire une adaptation à des phénomènes *endogènes* à l'exécution des programmes :
  - consommation mémoire,
  - appels de procédures (nombre, paramètres fréquents, ...),
  - « poids » dynamiques respectifs des processus sur le processeur,
  - services du système d'exploitation disponibles, etc.
- Avec les systèmes embarqués ou communiquants, elle s'est intéressée de plus en plus à de l'adaptation à des phénomènes *exogènes*, c'est-à-dire de l'environnement :
  - variation de bande passante des réseaux sans fil, la mobilité, la consommation d'énergie,
  - performances des services distants utilisés, etc.

Lorsque l'environnement physique (réel) est impliqué, on doit parler de *systèmes cyber-physiques*.

Auto-adaptabilité et théorie des systèmes II

- Pour *modéliser correctement* les phénomènes exogènes, mais aussi les phénomènes endogènes quantitatifs (performances, etc.), il faut comprendre leur *nature*, leur *évolution* et leur *impact* sur l'exécution.
- Mais tout aussi important, il faut comprendre l'effet des adaptations des applications sur ces phénomènes et en particulier leurs écolutions en continu.
- Ce sont des thèmes qui ont été étudiés dans les domaines de la *théorie des systèmes* et de la *théorie du contrôle*, et pour lesquels des modèles ont été proposés.
- Nous allons maintenant explorer les contributions de la *théorie des systèmes*, non pour en devenir des spécialistes, mais pour en connaître les principaux concepts et être capable d'en comprendre les implications.

Les contributions de la théorie du contrôle seront vues au prochain cours.

Qu'est-ce qu'un système et pourquoi les étudier ?

- La science et l'ingénierie s'intéressent depuis longtemps à la notion de système, la première pour mieux les *comprendre*, l'autre pour mieux les *construire*.
- La notion de *système* est d'autant plus difficile à cerner qu'elle a été étudiée sous différents angles et dans différents contextes :
  - systèmes *naturels*, comme le système gravitationnel du soleil, de ses planètes et de leurs satellites ;
  - systèmes *sociaux*, comme la ville, avec ses habitants, ses activités, ses transports, etc. ;
  - systèmes *technologiques*, comme un vaisseau spatial, son fonctionnement mécanique et électronique, ses instruments de pilotage, ses instruments scientifiques, etc.
- L'une des principales raisons d'étudier les systèmes est de savoir les *contrôler*, c'est-à-dire agir sur certaines de leurs propriétés, sur lesquelles on a prise, de manière à en influencer d'autres (performance, consommation, etc.) qui ont un intérêt particulier.

Fonctions et finalités des systèmes

- On attribue généralement à un système une *fonction*, intentionnelle ou non, qui explique son comportement et associe ses différentes propriétés.
- À cette fonction, on prête des *qualités* qui vont donner lieu à une *évaluation*, ce qui conduit, lorsque c'est possible, à la recherche des *meilleures solutions*, celles présentant les meilleures qualités, la meilleure évaluation.
  $\Rightarrow$ *En informatique, la qualité de service par exemple...*
- Ainsi, pour *comprendre*, *analyser* et *mieux construire* un système, il est très utile de le *modéliser* de manière à *capturer* ses propriétés essentielles, *inférer* les qualités sur ce modèle à partir de ses propriétés et ainsi *prévoir* le plus fidèlement possible le comportement du système réel.

Nature et propriétés des systèmes

- Les systèmes informatiques exhibent des propriétés dont les évolutions sont liées à des *phénomènes discrets* vus comme des *événements* se produisant à des *instants ponctuels* dans le temps :
  - recevoir d'une requête,
  - appeler d'un service,
  - presser un bouton, etc.,

  $\Rightarrow$ Des phénomènes modélisés par des *automates* changeant d'état aux occurrences d'*événements*.
- Mais ils exhibent également des propriétés dont les évolutions *graduelles* sont mieux capturées par des *variables continues* :
  - la durée d'un calcul,
  - le nombre moyen de requêtes par seconde,
  - la consommation d'énergie, etc.

  $\Rightarrow$ Des phénomènes nécessitant des *équations* mathématiques *continues*.

Les systèmes hybrides

- Les systèmes qui exhibent *à la fois* des évolutions *continues* et *discrètes* sont dits ***hybrides***.
- C'est le cas des systèmes cyber-physiques dont la partie informatique implique des évolutions discrètes et dont certaines propriétés internes quantitatives de même que plusieurs des propriétés physiques exhibent des évolutions continues.
- Le terme « *systèmes hybrides* » désigne une famille de modèles mathématiques, formels donc, étudiés depuis 25 ans environ, par quatre communautés de recherche :
  - mathématiques,
  - informatique théorique,
  - automatique (théorie du contrôle) et
  - recherche opérationnelle et théorie de la décision.
- Avant d'explorer les systèmes hybrides, rappelons les concepts essentiels de la modélisation des systèmes continus.

Modélisation par entrées/sorties des systèmes *continus*

- Considérons un ensemble de propriétés continues, modélisées par un ensemble de variables continues $v_1, \ldots, v_n$ et leurs domaines de valeurs comme les nombres réels.
- L'approche de modélisation par entrées/sorties demande d'abord de considérer une partition des $\{v_i\}, i = 1, \ldots, n$:
  - les variables *en entrée* $u_1, \ldots, u_p$ dont les valeurs sont dites *contrôlées* (ex. : mémoire allouée, nombre de MV, ...), et
  - les variables *en sortie* $y_1, \ldots, y_m$ dont les valeurs sont *observées* (ex. : temps moyen de traitement des requêtes).
- Prises dans un intervalle de temps $[t_0, t_f]$, chacune de ces variables évoluent continûment en prenant différentes valeurs (contrôlées ou observées) qui vont ainsi définir des fonctions :

$$\{u_1(t), \ldots, u_p(t)\} \qquad t_0 \leq t \leq t_f$$
$$\{y_1(t), \ldots, y_m(t)\} \qquad t_0 \leq t \leq t_f$$

Ce qu'on appelle également des *signaux*.

Comportement du système

- L'étude formelle d'un système continu capture son évolution, son *comportement*, par des fonctions mathématiques décrivant ses sorties selon de ses entrées par un ensemble d'équations :

$$\begin{aligned} y_1(t) &= g_1(u_1(t),...,u_p(t),t) \\ &... \\ y_m(t) &= g_m(u_1(t),...,u_p(t),t) \end{aligned}$$

  ou, dans une notation vectorielle

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{u}(t),t)$$

- Cette forme de modélisation a été étudiée intensivement et développée pour décrire formellement l'évolution continue des systèmes déterministes : mécaniques, électriques, pneumatiques, chimiques, biologiques...

La notion d'état

- Le principe selon lequel un système pourrait être totalement décrit par une fonction entre ses entrées et ses sorties est (le plus) souvent une simplification de la réalité.
- La plupart des systèmes exhibent une « mémoire » de leur évolution passée qui influe sur les sorties pour une même entrée.
- Cette mémoire peut être représentée par un ensemble de variables $\mathbf{x} = x_1,...,x_k$ dont les valeurs $\mathbf{x}(t)$ à un instant donné dénotent un *état* du système.
- *Comment modéliser la dépendance du système et de son état envers son évolution passée ?* Il est logique de supposer que :
  - l'état $\mathbf{x}(t)$ dépend de l'état initial $\mathbf{x}(0) = \mathbf{x}_0$ du système et des entrées $\mathbf{u}(s), 0 \leq s \leq t$ depuis le début du fonctionnement et que
  - les sorties $\mathbf{y}(t)$ dépendent de l'état courant $\mathbf{x}(t)$ et des entrées courantes $\mathbf{u}(t)$, l'état courant résumant donc tout le passé.

Équations modélisant la dynamique de l'état

Équations d'état

Dans un modèle, l'ensemble des équations requises pour spécifier l'état $\mathbf{x}(t)$ pour tout $t \geq t_0$, étant donnés l'état initial $\mathbf{x}_0$ et une fonction $\mathbf{u}(t), t \geq t_0$, des entrées, sont appelées *équations d'états*.

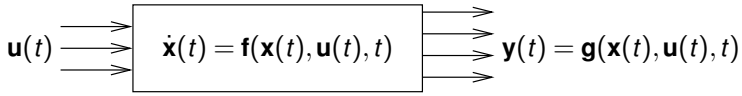
- Les équations d'états peuvent prendre de nombreuses formes, mais les systèmes continus ont été souvent étudiés sous l'angle des *équations différentielles* de la forme :

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t)$$

- Un modèle de système à espace d'états est alors défini par le système d'équations :

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t) \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t),t) \end{aligned}$$

Schéma de modélisation par espace d'états



Où :

- $\mathbf{u}(t)$: entrées externes (incluant le contrôle volontaire et les influences externes non-contrôlées) ;
- $\dot{\mathbf{x}}(t)$: dérivée *i.e.*, taux, de l'évolution de l'état ;
- $\mathbf{x}(t)$: évolution de l'état ;
- $\mathbf{y}(t)$: fonction (ou signal) de sortie.

Remarques

- La modélisation par espace d'états propose une approche élégante pour comprendre les systèmes, mais soulève deux difficultés centrales :
  - exprimer précisément le comportement d'un système par un ensemble de variables et d'équations *conformes à la réalité*, et
  - résoudre ces systèmes d'équations de manière à pouvoir calculer et prédire de manière sûre, grâce au modèle (*i.e.*, obtenir la fonction $\mathbf{y}(t)$), le comportement du système *réel*.
- De nombreux systèmes réels doivent être modélisés par des équations différentielles si complexes que les mathématiciens n'arrivent pas à résoudre *analytiquement*.
- Deux possibilités s'offrent alors :
  1. la simulation numérique du modèle pas-à-pas sur $[t_0, t_f]$, ou
  2. la simplification du modèle pour retomber sur des équations qu'on saura résoudre analytiquement dont, par exemple, l'approximation *linéaire* qui a été beaucoup étudiée et utilisée en pratique.

Systèmes hybrides stochastiques

- La plupart des systèmes réalistes ne sont pas *déterministes*.
  - Beaucoup exhibent des *variations aléatoires* plus ou moins limitées autour d'un comportement principalement déterministe que l'on peut assimiler à des erreurs comme par exemple, du jeu dans une mécanique.
  - D'autres exhibent des *comportements aléatoires* intrinsèques comme par exemple le délai entre les arrivées des clients à un guichet ou encore la variation de la bande passante d'un réseau sans fil.
- Capturer ces phénomènes exige des *modèles stochastiques*.

Systèmes hybrides stochastiques

Systèmes hybrides dont certains comportements peuvent se produire à des moments aléatoires en temps continu, et causer des évolutions d'états qui peuvent être modélisées de manière stochastique.

La notion de contrôle d'un système

- Puisque certaines variables en entrée peuvent être volontairement (arbitrairement) fixées, il est possible d'exercer un *contrôle* du système par le *choix* des valeurs de ces variables.
- Mais pouvoir fixer volontairement les valeurs de ces variables ne dit rien sur le choix à faire.
  Un critère de choix doit être utilisé !
- Conceptuellement, il s'agit de les choisir de manière à obtenir des sorties jugées *préférables*, notion restant à préciser.
- La détermination d'une *loi de contrôle* permettant d'obtenir ces sorties préférables est le sujet central en automatique et une partie du domaine de la décision appelée le *contrôle optimal*.
- Rappel : le contrôle sera le sujet du prochain cours.

Plan

1 Théorie des systèmes
2 Systèmes hybrides
3 Modélisation hybride d'un système auto-adaptable
4 Panorama des modèles hybrides
5 Première décomposition du modèle de l'exemple
6 Modélisation des systèmes auto-adaptables

Modèles de comportement discrets

- En informatique théorique et en programmation concurrente, le comportement des systèmes a été étudié en considérant des *espaces d'états* et de *transitions discrets*.
- Les modèles discrets développés à cet effet sont les *automates*, déclinés dans de nombreuses formes (AEF, déterministes, non-déterministes ou probabilistes ; réseaux de Petri ; ...).
- Ces modèles sont ensuite utilisés pour étudier le comportement puis vérifier les systèmes (*model-checking*, par exemple), c'est-à-dire en démontrer des propriétés comme la vivacité, l'atteignabilité de certains états, etc.
- Lorsque ces modèles sont appliqués à des systèmes qui ont des composantes continues ou discrètes mais infinies, ils forcent une discrétisation et une restriction à des ensembles finis qui ne capturent pas toujours les propriétés essentielles du système.

Vers des modèles comportementaux hybrides

- Ainsi, les modèles à base d'automates ne prennent en compte que le comportement par événements discrets se produisant en temps discret sur des espaces d'états discrets et finis.
- Les systèmes auxquels nous sommes confrontés ont aussi des comportements en temps continu, pouvant être modélisés par un espace d'états continu soumis :
  - à des *transitions* par événements définis par des paramètres discrets *et* continus et se produisant en temps *continu* mais à des instants *ponctuels*[1] ;
  - à des *évolutions* en temps continu pouvant être décrites par les équations continues de la théorie des systèmes classique.
- Une modélisation complète de ce type de systèmes exige des approches combinant *événements* (ponctuels mais à paramètres continus) en temps continu avec des *équations continues*.

[1] donc bien une notion d'événement *différente* de celle de modèles discrets car n'étant plus définis par des espaces finis.

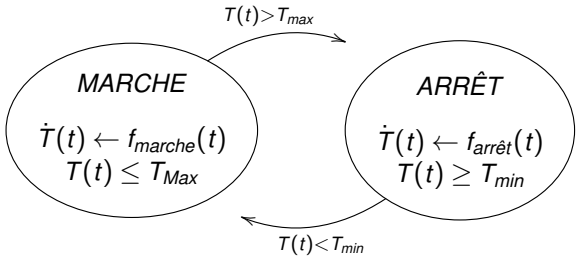
Théorie des systèmes hybrides

Définition

La théorie des système hybrides étudie le comportement des systèmes et les formalismes de modélisation qui impliquent à la fois des évolutions continues sur des espaces continus, décrites par des équations continues, et des transitions discrètes ou ponctuelles en temps continu décrites par des automates à états.

- Objectif affiché : analyser, vérifier et optimiser selon des modèles rigoureux les systèmes de contrôle cyber-physiques et les systèmes autonomiques.

Exemple classique de système hybride : thermostat



- Deux modes de fonctionnement : *MARCHE* (chauffe) ou *ARRÊT*.
- Deux seuils : T_{max} pour passer de l'état *MARCHE* à l'état *ARRÊT*, et T_{min} pour l'inverse (inertie ou *hystérésis*).
- Deux systèmes d'équations différentielles modélisent l'évolution de la température selon le mode courant de fonctionnement.

Observations

- La nature hybride du système de chauffage thermostatique vient du fait qu'il possède
  - un espace d'états continu, la température ambiante,
  - et un espace d'états discret, son mode de fonctionnement,

  qui *s'influencent l'un l'autre*.
- Le comportement continu (température) dépend du comportement discret (mode de fonctionnement) car l'évolution de la température dépend du fait qu'on chauffe (marche) ou non (arrêt).
- Le comportement discret (transition de modes) résulte du comportement continu (température), dans la mesure où le basculement d'un mode à l'autre dépend de la température ambiante par les deux seuils T_{min} et T_{max}.
  - (Note : ici, ces seuils sont liés aux décisions mais dans d'autres cas, ils peuvent aussi apparaître naturellement ; par exemple, une cuve qui déborde...).

Interactions continu/discret

- Vision hybride : systèmes dont l'évolution continue est interrompue par des phénomènes discrets dont l'impact est d'abord de changer l'état discret mais aussi l'état et le comportement continu (entrées, sorties, état, équations).
- Quatre types de phénomènes discrets [Branicky, 2005] :
  1. le *saut autonome*, lorsqu'un système change ponctuellement d'*état* de manière endogène ;
  2. la *commutation autonome*, lorsqu'un système change ponctuellement de *mode* de manière endogène pour adopter un nouveau modèle d'évolution (nouvelles équations) ;
  3. la *saut contrôlé*, lorsqu'un système change ponctuellement d'*état* suite à une modification d'une variable d'entrée (contrôle) ; et,
  4. la *commutation contrôlée*, lorsque le système change ponctuellement de *mode* suite à une modification une variable d'entrée (contrôle) pour adopter un nouveau modèle d'évolution (nouvelles équations).

Plan

1 Théorie des systèmes
2 Systèmes hybrides
3 Modélisation hybride d'un système auto-adaptable
4 Panorama des modèles hybrides
5 Première décomposition du modèle de l'exemple
6 Modélisation des systèmes auto-adaptables

Exemple fil rouge : Molène[1]

- Composants échangeant des données via réseau WiFi :
  - deux composants échangent des données lourdes, dont l'un sur un PC et donc sur batterie ;
  - possibilité de compresser les données et les décompresser pour accélérer les échanges sous bande passante faible ;
  - mais compresser coûte de la batterie car cela nécessite du calcul, donc si on peut éviter...
- Objectif de l'adaptabilité :
  - Améliorer le taux de transfert des données (incluant compression et décompression) tout en maintenant une durée de fonctionnement du PC sur batterie la plus longue possible.
- Propriétés adaptables ou influençables :
  - taux de transfert des données en $Mbits/s$
  - taux d'attrition de la batterie en mAh/s

[1] Inspiré de la thèse de Maria-Teresa Segarra, IRISA/Rennes I, 2000.

Éléments de modélisation I

- Données du modèle :
  - p : bande passante du réseau sans fil en $Mbits/s$, $0 \leq p \leq P_{max}$, avec interruptions aléatoires.
  - ΔB_{NC} (resp. ΔB_C, ΔB_0) : taux d'attrition de la batterie sans compression (resp. avec compression, sans réseau) en mAh/s
  - v_c : vitesse de compression en $Mbits/s$
  - v_d : vitesse de décompression en $Mbits/s$
  - τ_c : taux de compression moyen des données, $0 < \tau_c < 1$
- Évolutions contrôlées :
  1. taux de transfert des données τ_c selon le mode choisi (avec ou sans compression).
  2. consommation batterie supposée déterministe selon le mode :
     $$\dot{b}(t) = -\Delta B_{NC} \qquad \dot{b}(t) = -\Delta B_C \qquad \dot{b}(t) = -\Delta B_0$$
- Évolutions non-contrôlées :

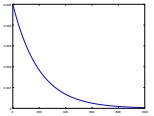
Éléments de modélisation II

1. bande passante $\Rightarrow$ ED stochastique (mouvement brownien).

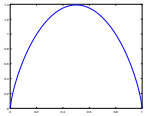
$$\dot{p}(t) = \sigma(p(t))d\mathcal{P}(t)$$
$$d\mathcal{P}(t) \sim \textbf{Exp}[\lambda_p] \in [0,\infty[\qquad \lambda_p = \text{dérivée moyenne}$$
$$\sigma(p) = \begin{cases} -1 & \textbf{if } u < p/P_{max} \\ 0 & \textbf{if } u = p/P_{max} \\ 1 & \textbf{if } u > p/P_{max} \end{cases} \qquad \text{avec } u \sim \textbf{U}[0,1] \in [0,1]$$

2. interruptions du réseau : modèle de panne

$$\text{durée entre les interruptions} = x_1 \sim \textbf{Exp}[\lambda_1]$$
$$\text{durée des interruptions} = x_2 \sim \textbf{Exp}[\lambda_2]$$
$$\text{bande passante à la reprise} = rP_{max} \text{ où } r \sim \textbf{Beta}[\alpha_p, \beta_p] \in [0,1]$$



Exp[200]



Beta[1.75, 1.75]

Adaptations

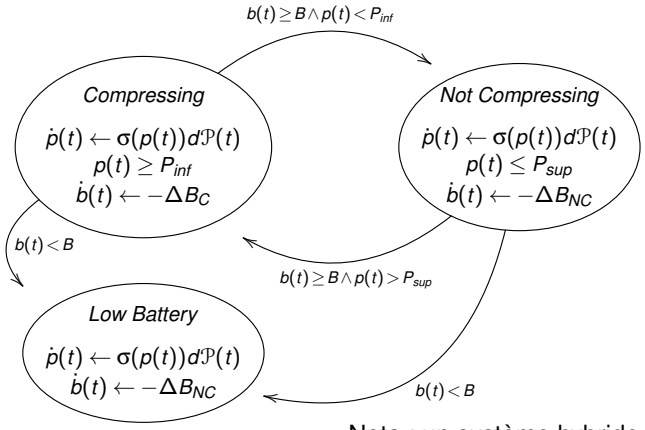
- Trois opérations d'adaptation :
  1. Mise en place de la compression.
  2. Retrait de la compression.
  3. Passage en mode « batterie faible » (compression interdite).
  - Intuitivement, il existe une BP seuil au-dessus de laquelle le taux de transfert est plus élevé sans compression mais en-dessus de laquelle il est plus élevé avec.
- Choix de politique d'adaptation (loi de contrôle) :
  - Politique à seuils avec hystérésis ($P_{inf} < P_{sup}$) pour la compression :
    - $p \geq P_{sup} > p_s$: suppression de la compression
    - $p \leq P_{inf} < p_s$: mise en place de la compression
    - ex.: $P_{sup} = 25$ Mbits/s et $P_{inf} = 21$ Mbits/s
  - Politique à seuil B pour la batterie :
    le niveau baissant continument, pas besoin d'hystérésis
    - $B < b \leq B_{max}$: autoriser la compression
    - $b \leq B < B_{max}$: ne plus autoriser la compression
    - ex.: $B_{max} = 5000$ mAh, seuil $B = 2000$ mAh

Estimation du seuil de BP par le taux de transfert

- Soit K *Mbits* de données à transmettre, la durée d nécessaire est :
$$d = \begin{cases} \frac{K}{p} & \neg \textit{ compression} \\ \frac{K}{v_c} + \frac{\tau_c K}{p} + \frac{\tau_c K}{v_d} & \textit{compression} \end{cases}$$
- Quelques manipulations donnent le taux de transfert :
$$\tau_t = \begin{cases} p & \neg \textit{ compression} \\ \frac{v_c p v_d}{p v_d + \tau_c v_c v_d + \tau_c p v_c} & \textit{compression} \end{cases}$$
- Et le seuil p_s justifiant le passage d'un mode à l'autre :
$$p_s = \frac{(1-\tau_c) v_c v_d}{v_d + \tau_c v_c}$$

Ex.: pour $v_c = 50\,Mbits/s$, $\tau_c = 0.4$, $v_d = 75\,Mbits/s$, on trouve $p_s \approx 23.68\,Mbits/s$

Présentation graphique (sans interruptions réseau)



Nota : un système hybride monolithique « incluant » le contrôle dans ses conditions de transition.

Plan

1 Théorie des systèmes
2 Systèmes hybrides
3 Modélisation hybride d'un système auto-adaptable
4 Panorama des modèles hybrides
5 Première décomposition du modèle de l'exemple
6 Modélisation des systèmes auto-adaptables

Un domaine de recherche en pleine ébullition

- Sur une même base conceptuelle, pléthore de choix possibles.
- Donc, foison de modèles.
- État actuel de la recherche : exploration des possibles.
- Quatre principales approches ou communautés de recherche :
  - informatique théorique : automate hybride (Henzinger, Lynch), vivacité, composition ;
  - mathématique : garanties sur l'existence de solutions, vers des formulations uniformisant évolutions discrètes et continues ;
  - automatique : contrôle continu/discret, couplage entre deux systèmes hybrides, contrôlé (*plant*) et contrôleur ;
  - décision : synthèse de contrôleur optimaux, modèles stochastiques, processus de décisions markoviens (PDM).

Voic un panorama succint, avec une orientation architectures logicielles !

Automates hybrides d'Henzinger

Travaux remontant au milieu des années '90 :

- espace d'états incluant des variables continues,
- événements (appelés actions) discrets et évolutions continues,
- ensembles de variables homogènes entre états « discrets »,
- description du comportement par traces alternant transitions discrètes et trajectoires continues,
- pas de restriction sur la manière de définir les trajectoires continues (équations, équations différentielles, . . .),
- composition par partage d'événements et de variables,
- notion de vivacité en continu (appelée réceptivité).

Familles d'automates hybrides E/S de Lynch et al.

- Hybrid I/O Automata (mi '90 $\rightarrow$ 2003) :
  - partition des événements et des variables : entrées externes, sorties externes, internes ;
  - composition : connexion d'événements et de variables en sortie avec des événements et variables en entrée.
- Timed I/O Automata (post-HIOA, $\neq$ timed automata)
  - pas de communication par connexion de variables continues, seulement par connexion d'événements mais pouvant inclure des valeurs ponctuelles de variables continues ;
  - tient compte de la nature discrète des échanges par réseau.
- Vision conception logicielle/système :
  - HIOA : conception modulaire d'artéfacts monolithiques centralisés.
  - TIOA : conception modulaire d'artéfacts décentralisés.
  - Analogie avec les GALS : composition de TIOA formés par composition d'HIOA où toutes les variables continues sont internes.

Contrôle des systèmes hybrides

- Notion de contrôle dans le modèle :
  - les entrées $u(t)$ sont vues comme un signal de contrôle continu sur les équations (comme on l'a vu),
  - des événements émis volontairement (actions) et provoquant des transitions discrètes impulsent un contrôle discret.
- Mise en place d'une loi de contrôle, deux possibilités :
  - fondre la loi de contrôle directement dans le modèle (exemple thermostat), ou
  - composition parallèle d'un système hybride contrôleur proposant le contrôle en sortie sous forme d'événements ou de fonctions continues, et d'un système hybride contrôlé (*plant*) recevant le contrôle en entrée.
- Séparer ouvre la possibilité d'utiliser des modèles différents entre le contrôleur et le contrôlé.
- On y reviendra un peu plus loin...

Puissance de modélisation des systèmes hybrides

- Informatiquement parlant, un système hybride est un aussi *modèle de calcul*, comme n'importe quel automate, produisant un état du système à tout instant t.
- En termes de systèmes auto-adaptables, un système hybride peut être défini pour exprimer :
  - le comportement du système à contrôler,
  - le calcul réalisé par les capteurs pour *échantillonner* et *filtrer* les données,
  - le calcul de la valeur de la loi de commande par un contrôleur,
  - l'échantillonnage discret réalisé par la communication des valeurs des variables continues à des instants discrets,
  - le délai de communication de valeurs discrètes via le réseau, etc.
- L'utilisation des HIOA et TIOA, avec composition par partage de variables et d'actions discrètes, permet également de structurer les systèmes hybrides en sous-systèmes et plus généralement de modéliser les systèmes hybrides de *contrôle en réseau*, les communications distantes se faisant entre TIOA.

Plan

1 Théorie des systèmes
2 Systèmes hybrides
3 Modélisation hybride d'un système auto-adaptable
4 Panorama des modèles hybrides
5 Première décomposition du modèle de l'exemple
6 Modélisation des systèmes auto-adaptables

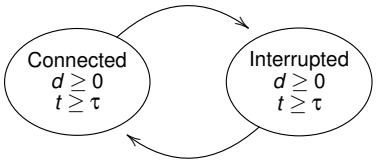
Modélisation modulaire

- Modélisation précédente : monolithique, peut rapidement devenir complexe, peu réutilisable.
- Passage à une modélisation modulaire, plus fidèle à l'architecture logicielle, en utilisant l'approche HIOA/TIOA.
- Illustration sur l'exemple :
  - Modèle de bande passante exportant des événements d'interruption et de reprise et la variable p.
  - Modèles de contrôleur côté PC et serveur qui importent les variables p et b et émettent des événements de contrôle forçant le passage à la compression, le retrait de la compression ou le mode batterie faible.
  - Modèle du PC émettant la variable b et important les événements de contrôle.
  - Modèle du serveur important les événements de contrôle.

On va se concentrer sur le premier...

Bande passante : modèle des interruptions du réseau

when $d{=}t{-}\tau$; **emit** {Interrupt}, **reset** $d \leftarrow x_2 \sim \mathbf{Exp}[\lambda_1], \tau \leftarrow t$



when $d{=}t{-}\tau$; **emit** {Resume}, **reset** $d \leftarrow x_1 \sim \mathbf{Exp}[\lambda_2], \tau \leftarrow t$

t représente le temps.

Importe	Exporte
	Interrupt Resume

où :

$\lambda_1 =$ durée moyenne des interruptions

$\lambda_2 =$ temps moyen entre les interruptions

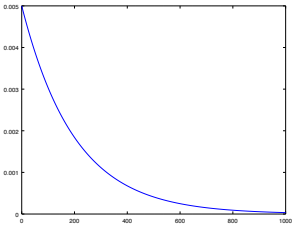
$t_0 = 0$

$d_0 = 0$

$\tau_0 = 0$

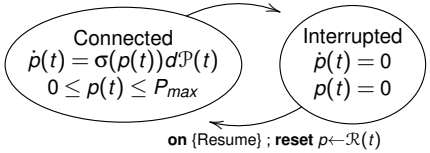
$s_0 = (\text{Interrupted}, d_0, \tau_0)$

Exp[200]



Bande passante : évolution continue

on {Interrupt} ; **reset** $p \leftarrow 0$



on {Resume} ; **reset** $p \leftarrow \mathcal{R}(t)$

Importe	Exporte
Interrupt Resume	
	p

où :

$d\mathcal{P}(t) \sim \mathbf{Exp}[\lambda_p] \in [0, \infty[$

$\lambda_p =$ dérivée moyenne de la bande passante

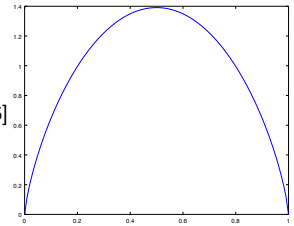
$$\sigma(p) = \begin{cases} -1 & \textbf{if } u < p/P_{max} \\ 0 & \textbf{if } u = p/P_{max} \\ 1 & \textbf{if } u > p/P_{max} \end{cases}$$

avec $u \sim \mathbf{U}[0,1] \in [0,1]$

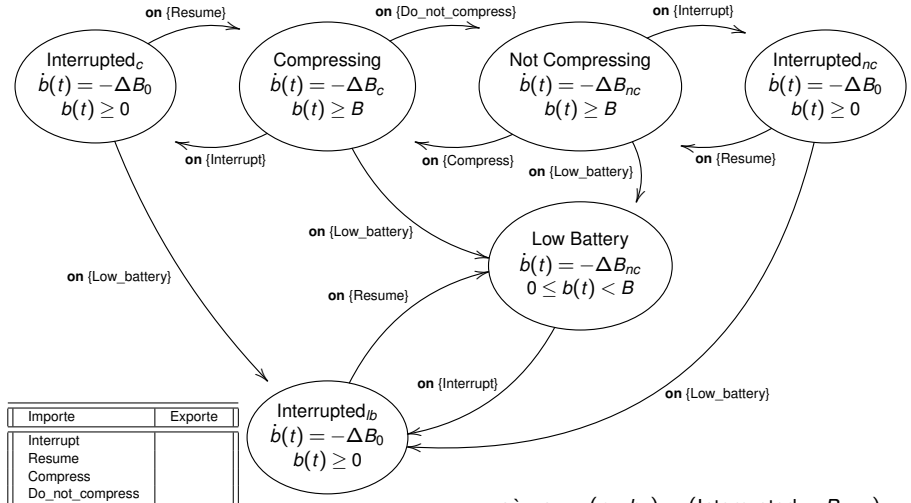
$\mathcal{R}(t) = rP_{max}$, avec $r \sim \mathbf{Beta}[\alpha_p, \beta_p] \in [0,1]$

$s_0 = (q_0, p_0) = (\text{Interrupted}, 0)$

Beta[1.75, 1.75]



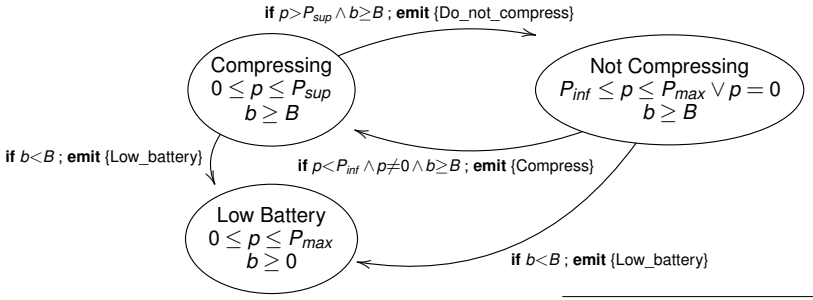
Modèle du PC : niveau de la batterie



Importe	Exporte
Interrupt Resume Compress Do_not_compress Low_battery	
	b

où : $s_0 = (q_0, b_0) = (\text{Interrupted}_{nc}, B_{max})$

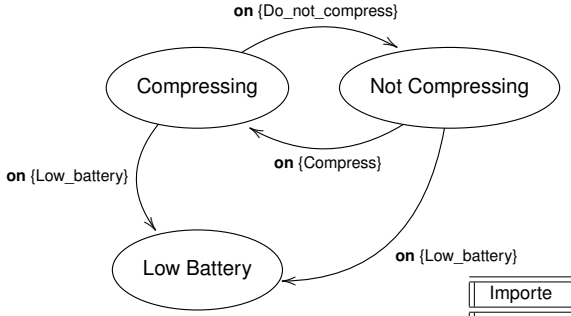
Modèle du contrôleur du PC (et côté serveur) : décisions



où $s_0 = (q_0, p_0, b_0) = (\text{Not Compressing}, p_0, b_0)$
avec p, b importées

Importe	Exporte
	Compress Do_not_compress Low_battery
p, b	

Modèle du serveur

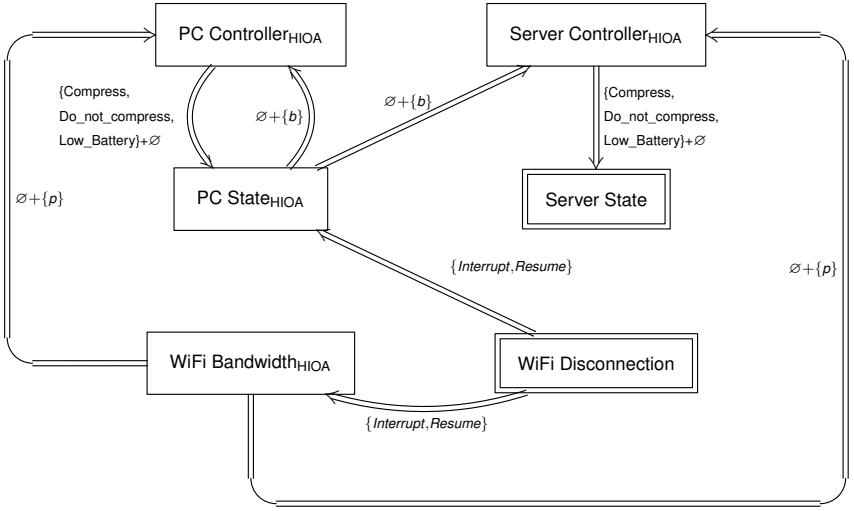


où $s_0 = q_0 = \text{Not Compressing}$

Importe	Exporte
Compress Do_not_compress Low_battery	

- En pratique, le serveur se contente mettre en place la compression ou la retirer.

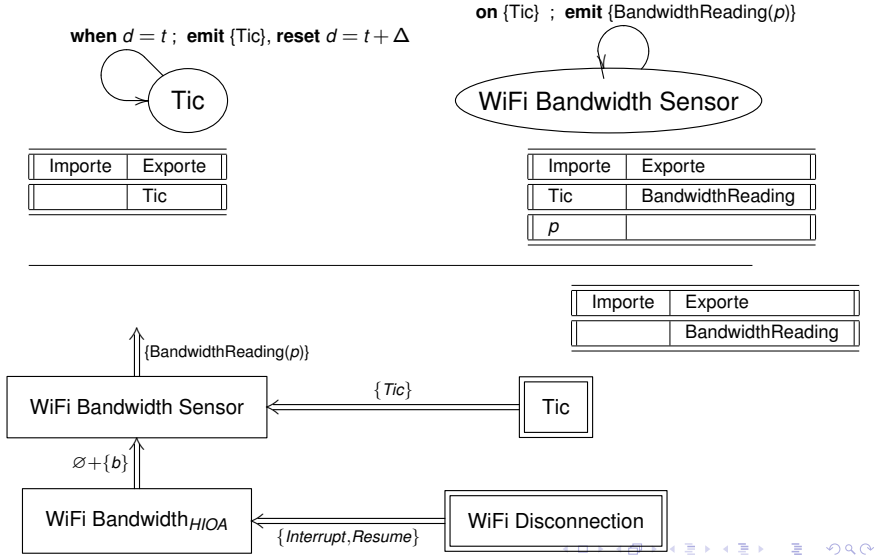
Composition du modèle complet (HIOA+TIOA)



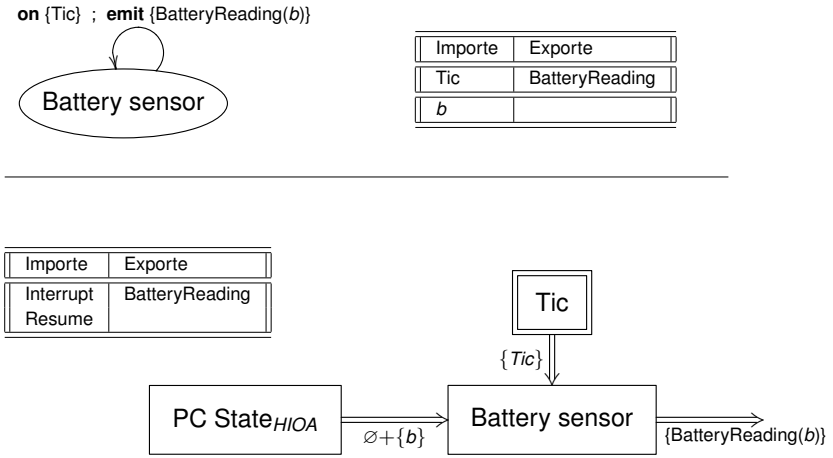
Réduction du couplage : vers un composition de TIOA

- La décomposition en HIOA implique le partage de variables continues et donc des modèles *fortement couplés, centralisés*.
- Pourquoi ? Observations :
  1. Modélisation irréaliste puisque l'achitecture logicielle correspondante *doit* être répartie entre le portable et le serveur.
  2. Or, aucun procédé numérique ne permet à un contrôleur informatique d'accéder *en continu* à la bande passante : il y aura nécessairement *échantillonnage* discret.
- Comment découpler les modèles ?
  - Former des TIOA en introduisant des modèles « échantillonneurs » : en quelque sorte des modèles de *capteurs*.
  - Ils s'interposent pour « lire » les valeurs des variables continues b, p et fournir des événements BandwidthReading et BatteryReading portant des valeurs *ponctuelles*.

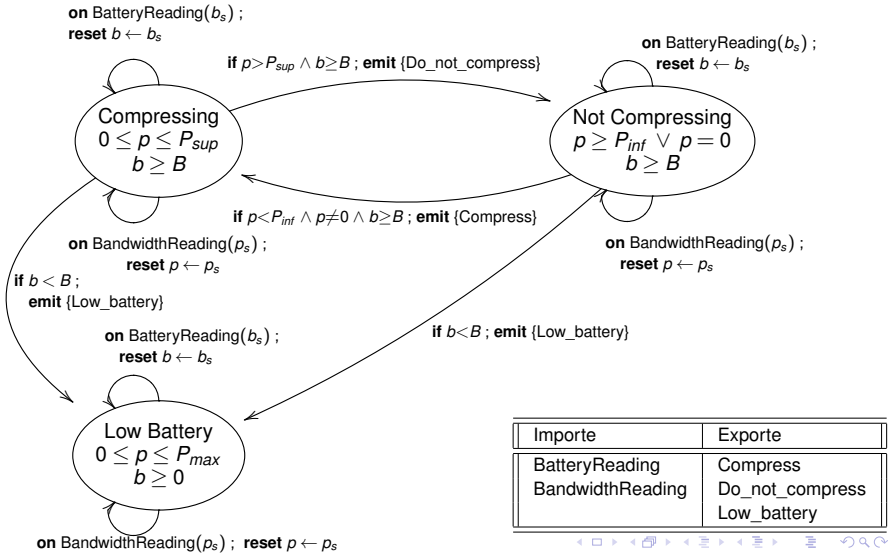
Modèle TIOA de de bande passante



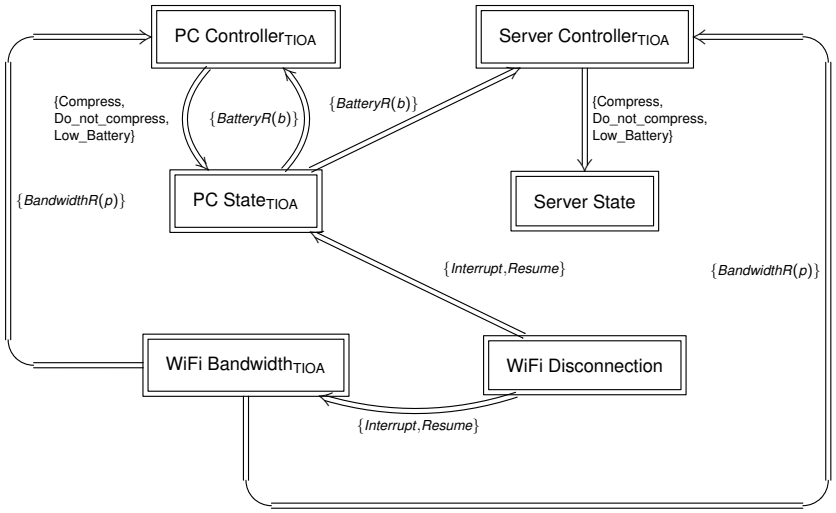
Modèle TIOA du PC



Modèle TIOA du contrôleur du PC



Modèle complet décomposé en TIOA uniquement



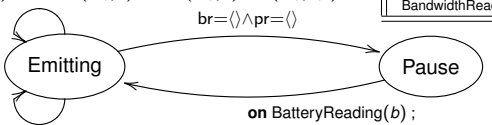
Introduction de la communication par réseau

- Le modèle précédent satisfait le besoin de découplage, mais n'est toujours pas réaliste.
- S'il y a un contrôleur côté serveur, les événements BatteryReading(b) doivent transiter par le réseau et donc être retardés par un certain délai de transmission.
- Si la bande passante n'est mesurée que d'un côté (cohérence), alors les événements BandwidthReading(p) devront aussi être transmis via le réseau.
- Comment modéliser cela ?
  $\Rightarrow$ Par un modèle recevant des événements puis les réémettant après un délai aléatoire.
- Illustre le potentiel des modèles hybrides (++) qui sont des modèles de calcul à part entière.

Modèle de transmission réseau

Importe	Exporte
BatteryReading BandwidthReading	BatteryReading BandwidthReading

when $t = e_t(\mathrm{br},\mathrm{pr})$; **emit** $e_v(\mathrm{br},\mathrm{pr})$, **reset** $(\mathrm{br},\mathrm{pr}) \leftarrow r(\mathrm{br},\mathrm{pr},t)$



on BatteryReading(b) ;
  reset br $\leftarrow (t+\tau,b)\S$ br $\mid \tau \sim$ **Gamma**(κ,θ)
on BandwidthReading(p) ;
  reset pr $\leftarrow (t+\tau,p)\S$ pr $\mid \tau \sim$ **Gamma**(κ,θ)

on BatteryReading(b) ;
  reset br $\leftarrow \langle (t+\tau,b) \rangle \mid \tau \sim$ **Gamma**(κ,θ)
on BandwidthReading(p) ;
  reset pr $\leftarrow \langle (t+\tau,p) \rangle \mid \tau \sim$ **Gamma**(κ,θ)

where :

$$\mathrm{br}(t_0) = \langle\rangle \qquad \mathrm{pr}(t_0) = \langle\rangle$$

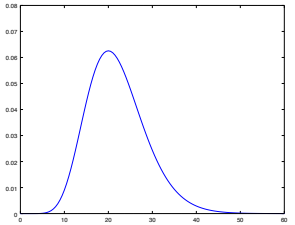
$$e_t(I_1,I_2) = \min\{x \mid (x,\cdot) \in I_1 \vee (x,\cdot) \in I_2\}$$

$$e_v(I_1,I_2,t) = \begin{cases} \mathrm{BatteryReading}(b) & \text{if } (t,b) \in I_1 \\ \mathrm{BandwidthReading}(p) & \text{if } (t,p) \in I_2 \end{cases}$$

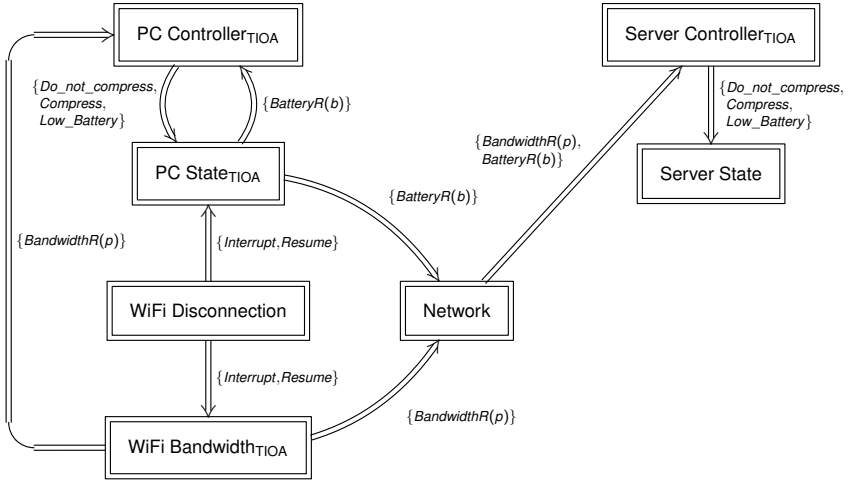
$$r_2(I_1,I_2,t) = \begin{cases} (r_1(I_1,t),I_2) & \text{if } (t,\cdot) \in I_1 \\ (I_1,r_1(I_2,t)) & \text{if } (t,\cdot) \in I_2 \end{cases}$$

$$r_1(I,t) = \begin{cases} \bot & \text{if } I = \langle\rangle \\ I_r & \text{if } I = (t,\cdot)\S I_r \\ (z,x)\S r_1(I_r,x) & \text{if } I = (z,x)\S I_r \wedge z \neq t \end{cases}$$

Gamma$[11,2]$
mean =
22 msec.



Modèle complet avec répartition explicite



Plan

1 Théorie des systèmes
2 Systèmes hybrides
3 Modélisation hybride d'un système auto-adaptable
4 Panorama des modèles hybrides
5 Première décomposition du modèle de l'exemple
6 Modélisation des systèmes auto-adaptables

Systèmes hybrides et informatique autonomique

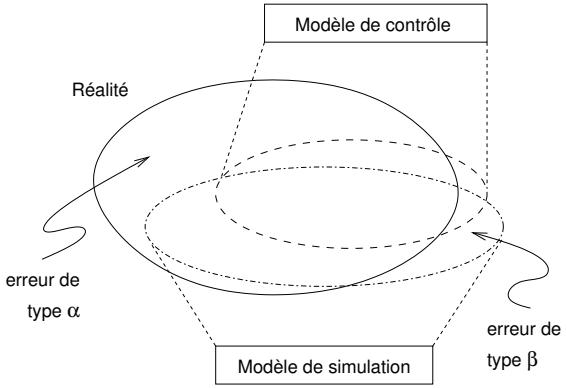
- De manière générale, la modélisation hybrides des systèmes permet de les comprendre, les analyser formellement et les construire, mais aussi de les simuler pour en étudier les trajectoires et leurs propriétés.
- Comment utiliser la modélisation hybride dans le développement des systèmes autonomiques ?
  1. Modéliser l'environnement et les éléments gérés de manière à les simuler pour les valider par une forme de test.
  2. Modéliser le contrôle pour calculer des politiques qui seront appliquées par les gestionnaires autonomiques, puis les simuler avec une politique donnée pour les valider.
- Plusieurs modèles hybrides peuvent devenir nécessaires :
  - *Un modèle complet et précis du système à contrôler est nécessaire pour faire des tests réalistes et fiables.*
  - *Mais pour des raisons de calculabilité des politiques de contrôle, le modèle de contrôle peut devoir être simplifié pour être résolu.*

Deux niveaux de modélisation : test et contrôle

- Il peut donc y avoir deux modèles, différents, co-existant dans la conception et la mise en œuvre d'un système autonomique :
  1. le modèle de comportement des éléments gérés et de l'environnement *dont fidélité est garante de la qualité de la validation du système autonomique*
  2. le modèle de contrôle pour les gestionnaires autonomiques *dont la fidélité est la clé d'une auto-adaptabilité dynamique correcte et efficace*
- Ainsi, le modèle de comportement de l'élément géré peut être volontairement plus complexe pour obtenir des simulations plus fidèles à la réalité, mais le modèle de contrôle volontairement plus simple pour faciliter le calcul d'une politique de contrôle.

Exemple : un système informatique peut être simulé comme un système hybride stochastique mais vu comme un système de contrôle linéaire déterministe pour appliquer des techniques bien connues de la théorie du contrôle.

Modèles hybrides pour l'informatique autonomique



Par analogie avec les tests statistiques, deux types d'erreurs de modélisation (modèle/réel, modèle/modèle) :

- α : comportement licite non capturé par le modèle,
- β : comportement inexistant en réalité mais exhibé par le modèle.

Récapitulons...

1. La ***théorie des systèmes*** s'intéresse à la modélisation des systèmes et en particulier de leur comportement dans le temps.
2. Les ***modèles quantitatifs***, à base d'états, permettent d'appréhender les variations dans les valeurs des propriétés des systèmes (comme la qualité de service) et donc d'introduire des possibilités de ***contrôle*** de leurs propriétés.
3. Les modèles quantitatifs classiques de la théorie des systèmes sont fondés sur des ***variables d'états continues***, des ***lois de contrôles continues*** et des évolutions exprimées par des ***équations différentielles***.
4. L'informatique et d'autres disciplines similaires ont aussi développé des modèles de ***systèmes discrets***, à base d'***automates***, d'états discrets et d'événements provoquant des transitions à des instants ponctuels, en temps continu ou discret.
5. Les systèmes ***cyber-physiques*** mélangent des aspects ***discrets et continus*** ; ils nécessitent des modèles capables de capturer tous ces phénomènes à la fois, en interaction : les ***systèmes hybrides***.
6. La modélisation des systèmes auto-adaptables et autonomiques fait intervenir ***deux niveaux*** : le ***modèle du système plutôt hybride***, pouvant servir à sa simulation, et le ***modèle de contrôle, hybride ou plus simple (continu ou discret)***, utilisé pour déterminer et appliquer les lois de contrôle pour l'auto-adaptabilité.

Pour aller plus loin : sélection de lectures recommandées

- *Introduction to Discrete-Event Systems*, C.G. Cassandras et S. Lafortune, Springer, 2008, chapitres 1 et 2.
  écrit pour des informaticiens, mais ne touche qu'aux systèmes discrets puis temporisés.
- *Introduction to hybrid systems*, W.P.M.H. Heemels, D. Lehmann, J. Lunze et B. De Schutter, chapitre 1 de *Handbook of Hybrid Systems Control — Theory, Tools, Applications*, Cambridge University Press, 2009.
  texte accessible, bien qu'écrit pour des mathématiciens.
- *Hybrid I/O Automata*, Nancy Lynch, Roberto Segala and Frits Vaandrager, Information and Computation 185, 2003, pages 105–157.
- *Timed I/O Automata : A Mathematical Framework for Modeling and Analyzing Real-Time Systems*, Dilsun K. Kaynar, Nancy Lynch, Roberto Segala and Frits Vaandrager, Proceedings of the 24th IEEE International Real-Time Systems Symposium, 2003, pages 166–177.
- *Stochastic Hybrid Systems Meet Software Components for Well-Founded Cyber-Physical Systems Software Architectures*, Jacques Malenfant. European Conference on Software Architecture (ECSA), September 9-13, 2019, Paris, France.