

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 7

De l'entité adaptable — principes et conception

Objectifs pédagogiques du cours 7

- Comprendre ce que représente en pratique une entité adaptable dans un système cyber-physique et dans la sous-classe des systèmes autonomiques.
- Comprendre les problématiques liées à la conception et l'implantation des éléments gérés *i.e.*, les entités adaptables des systèmes autonomiques.
- Établir des principes de conception et de modélisation qui vont servir de base à une méthodologie de développement logiciel spécifique aux entités adaptables.
- Comprendre les problématiques liées à la mesure des propriétés quantitatives dans les systèmes cyber-physiques, et développer la capacité à bien mesurer et à bien utiliser la mesure.
- Développer des savoir-faire dans la conception et la mise en œuvre des éléments gérés en informatique autonome.

Plan

- 1 Entités adaptables dans les systèmes cyber-physiques
- 2 Éléments gérés et leurs propriétés d'adaptation
- 3 Principes de conception pour l'adaptabilité
- 4 Conception logicielle des éléments gérés
- 5 Instrumentation, observation, mesure et calcul
- 6 Conseils pragmatiques pour la conception

Rappel : qu'est-ce qu'un CPS ?

- Un système cyber-physique en pratique peut être vu comme une extension d'un système embarqué :
 - il est constitué d'une partie *cyber*, un (ou un ensemble d') élément(s) de calcul (ordinateurs, micro-contrôleurs, ...) possiblement interconnectés par réseau ;
 - et il est en relation avec le monde *physique* réel avec lequel il interagit via des capteurs et des actionneurs.
- L'objectif de la mise en relation d'une partie cyber avec le monde physique est de pouvoir exercer un *contrôle* pour les adapter, contrôle dont les actions seront décidées et calculées par programme à partir des informations recueillies via les capteurs.
- Pour ce faire, on s'appuie sur la théorie du contrôle (introduite au cours 4) mais qu'il s'agit de réaliser en pratique, ce qui va poser des difficultés relevant de l'*ingénierie du contrôle*.
- La partie cyber est ici une entité de contrôle (contrôleur) dont la mise en œuvre sera étudiée un peu plus loin.

Qu'est-ce qu'une entité adaptable dans un CPS ?

- Dans un système cyber-physique au sens de l'embarqué, l'entité adaptable est immergée dans le monde réel :
 - cuve dans un procédé chimique dont l'alimentation et le vidage est contrôlé par des vannes, voiture autonome, drone, etc.
- Elle offre des capteurs et actionneurs qui sont des dispositifs mécatroniques en interface entre la partie cyber et le monde physique, permettant à la partie cyber d'obtenir des informations et d'exercer des actions sur le monde physique :
 - obtenir le niveau de liquide dans la cuve, ouvrir ou fermer les valves d'alimentation ;
 - obtenir la position relative de la voiture, sa vitesse courante ainsi que les obstacles autour d'elle, tourner le volant et/ou accélérer/freiner ;
 - obtenir la position du drone, sa vitesse courante, la vitesse du vent, augmenter ou réduire la propulsion, agir sur les gouvernes.

Et dans un système autonome ?

- Un système autonome est un CPS dont au moins une partie des entités adaptables sont en fait des entités « cyber » (programme, composants, intergiciels, etc.) ayant des propriétés similaires aux entités du monde physique.
- Pour être clair, adoptons la terminologie proposée par IBM et désigner ces entités adaptables « cyber » des *éléments gérés*.
- Dans l'esprit des systèmes cyber-physiques, les éléments gérés vont faire l'objet d'un contrôle, et pour ce faire :
 - ils vont être dotés de capteurs et d'actionneurs et
 - le contrôle pourra utiliser les mêmes techniques que dans un système cyber-physique classique.
- Ainsi, dans un système autonome, la partie cyber est constituée à la fois d'éléments gérés, et d'entités de contrôle.
 - La malléabilité du cyber permet de réappliquer ce schéma *récur-sivement* en rendant les entités de contrôle également adaptables, elles-mêmes adaptées par un *méta-contrôleur*.

Éléments gérés et architectures logicielles

- En termes d'architecture logicielle, un élément géré est une entité logicielle devant s'intégrer dans un système cyber-physique.
- Comme nous privilégions l'approche par composants, nous allons tendre vers des *composants cyber-physiques* intégrant les propriétés requises par les systèmes autonomes dont :
 - l'*exécution temps réel* : comme pour les système embarqués, pour mettre en œuvre un contrôle efficace et correct ;
 - l'*observation* : par des interfaces capteurs offertes, ports et connecteurs dédiés et les opérations de mesure correspondantes ;
 - l'*adaptation* : par des interfaces actionneurs offertes, ports et connecteurs dédiés et les opérations d'adaptation correspondantes qu'on sache exécuter sans impact négatif sur le bon fonctionnement de l'entité ;
 - la *spécification* et la *modélisation* : par des modèles formels et exécutables attachés au composant ;
 - les *interfaces riches* : exprimant les ressources requises et offertes par le composant et plus généralement l'ensemble de ses contraintes de fonctionnement.

Contrôlabilité

- Expliciter les propriétés adaptables, leur observabilité, leur visibilité et leur adaptabilité effective est central, mais insuffisant pour assurer leur *contrôlabilité* au sens décrit au cours 4.
- Un contrôle visant à amener ou maintenir les valeurs de propriétés dans un ensemble-cible, il est important aussi de s'assurer de l'effet des adaptations sur les propriétés et de leur capacité à atteindre cet ensemble-cible.

Principe de contrôlabilité

Toute **propriété influençable** par un ensemble d'**opérations** d'adaptation **visibles** et dont les **effets** sur la propriété sont **observables** et **visibles** doit également garantir qu'à partir de toute valeur initiale de la propriété, ces opérations peuvent ramener cette dernière **aussi près que souhaité** de l'objectif du contrôle dans un **temps fini**.

Propriétés qualitatives versus quantitatives

- Une hypothèse fondamentale faite sur les valeurs des propriétés est que l'on peut les *comparer* (égalité, inférieur, supérieur) et qu'on possède une notion de *distance* entre valeurs, ces distances pouvant aussi être *comparées* entre elles.
- Déjà, la physique et les sciences expérimentales nous ont appris que toute mesure est entachée d'*imprécision*, ce qui a un impact sur l'observabilité et la contrôlabilité.
- Dans certains cas, l'imprécision est telle que les propriétés sont mieux appréhendées de manière *qualitative*.
 - *Exemple : la température peut être chaude, tiède ou froide.*
- Le traitement des données qualitatives est un domaine de recherche actif qui a de nombreuses applications en pratique.
- Dans le domaine du contrôle, une approche qui a beaucoup de succès est la *logique floue* (nous y reviendrons).

Plan

- 1 Entités adaptables dans les systèmes cyber-physiques
- 2 Éléments gérés et leurs propriétés d'adaptation
- 3 Principes de conception pour l'adaptabilité
- 4 Conception logicielle des éléments gérés**
- 5 Instrumentation, observation, mesure et calcul
- 6 Conseils pragmatiques pour la conception

Conception ouverte ou fermée ? I

- L'approche de programmation par composants met l'accent sur leur conception indépendante de leur utilisation.
- Cela implique deux approches de conception des éléments gérés :
 - 1 **Conception ouverte** : la conception de l'élément géré précède celle de l'entité de contrôle ; il faut alors prévoir toutes les formes de contrôle qui pourraient s'appliquer pour intégrer les capteurs et actionneurs nécessaires.
 - 2 **Conception fermée** : la conception de l'entité de contrôle précède ou est concomitante à celle de l'élément géré, et alors on peut se contenter des seuls capteurs et actionneurs nécessaires et suffisants pour les besoins de l'entité de contrôle visée.
- Les informaticiens recherchent naturellement genericité et réutilisabilité, mais la théorie du contrôle montre que mettre en œuvre un contrôle correct et efficace introduit un *couplage très fort* entre les entités adaptables et de contrôle (cf. cours 4).

Cycle de production en instrumentation et mesure

- 1 Grandeur physique : propriété à mesurer et ses caractéristiques intrinsèques (type, unité de mesure, etc.).
- 2 Mesure : procédé par lequel la grandeur physique va être traduite en donnée utilisable.
- 3 Conditionnement : dans la durée, conversion des mesures en un signal reflétant l'évolution temporelle de la grandeur physique, tout en éliminant les biais systématiques dûs à la technique de mesure.
- 4 Traitement du signal : filtrage des erreurs statistiques de mesure.
- 5 Exploitation : consommation des données produites.

Ce cycle de production est proposé en physique et électronique, mais ces différents concepts trouvent un écho en informatique autonome également !

Propriétés et observation

- Dans un contexte informatique, observer suppose *mesurer* ou *calculer* la valeur d'une propriété, et donc
 - de programmer le code réalisant cette mesure ou ce calcul, et
 - de l'exécuter (répétitivement) à des instants discrets, c'est-à-dire procéder à un *échantillonnage*.
- Toute observation implique :
 - un certain *effort* ou *coût* pour l'obtenir,
 - une certaine *imprécision* ou *erreur* dans la mesure, et
 - une certaine *perturbation* du système (effet « quantique »).
- Observer suppose donc un compromis entre
 - les ressources consommées pour la mesure,
 - la précision de cette mesure et
 - l'ampleur de la perturbation du système.
- L'observation ne peut pas être vue comme une boîte noire masquée derrière des interfaces (capteurs) ; le *comment* est aussi important que le *quoi* !

Qu'est-ce qu'un capteur ?

- Dans le domaine de l'instrumentation et mesure, un capteur possède deux fonctions qui sont assez souvent confondues l'une avec l'autre mais qui n'ont pas les mêmes implications :
 - il *mesure ou calcule* la valeur d'une certaine propriété ;
 - il permet de *lire* cette mesure.

Exemple : considérons un thermomètre médical ; il est mis en place pour prendre la température, puis retiré ; la température s'affiche, par exemple sur un écran numérique, et elle peut être lue par une ou plusieurs personnes.

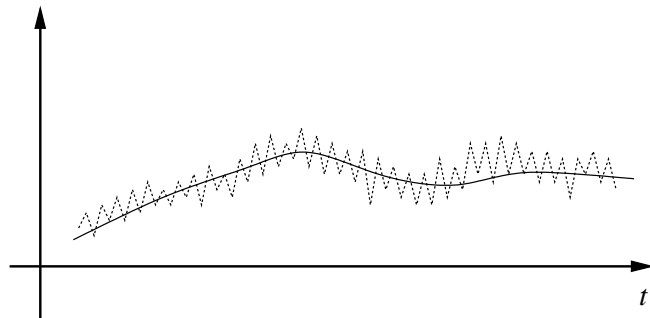
- Dans le contexte des entités adaptables,
 - la fonction de mesure/calcul se réalise par l'implantation de son code puis de son exécution, alors que
 - sa fonction de lecture se réalise par l'implantation de l'interface capteurs et ses appels par les clients de la valeur mesurée.

Lecture et mesure/calcul ne se confondent pas et ne sont pas nécessairement réalisés au même moment.

De l'évolution à la mesure temporelle : l'échantillonnage

- Une propriété p dont la valeur évolue dans le temps peut être modélisée par une fonction $p(t)$.
- L'évolution peut être continue ou discrète, et dans ce dernier cas elle peut être modélisée comme une fonction continue avec des paliers (en escalier), des pics de Dirac, ou des interpolations.
- Rappel : en informatique, toute mesure de l'évolution d'une propriété suppose de l'*échantillonnage* discret pour détecter les changements de valeurs et mesurer ces dernières.
- Échantillonner se décline à deux niveaux :
 - détecter, mesurer ou calculer à des instants discrets *i.e.*, *échantillonner pour mesurer* une propriété donnée grâce à l'algorithme implantant le capteur ;
 - lire la valeur à des instants discrets dans le temps *i.e.*, *échantillonner pour surveiller* la propriété via l'interface capteur la rendant visible.

- Pour un contrôleur utilisant p , les alias ne mènent pas aux mêmes décisions : par exemple $p(t)$ paraît croissante à t_0 alors que $p''(t)$ paraît décroissante.
- *Aucune fréquence* ne permet de garantir l'absence d'aliasage et attention aux sous- ou sur-échantillonnage :
 - Sous-échantillonnage : des variations transitoires peuvent passer inaperçues au contrôleur.
 - Sur-échantillonnage : augmentation du coût et risque de faire apparaître des variations qui ne sont dues qu'aux erreurs de mesure.
- *Problème : déterminer une fréquence d'échantillonnage* demande de bien connaître le phénomène échantillonné, son amplitude, son évolution ; à chaque fréquence correspond une borne sur la variation de la fonction, et donc ses dérivées, que l'échantillonnage pourra capturer.



- En pointillés, $p'(t)$ obtenue en reliant les mesures bruitées.
- Mène à surréagir à des variations insignifiantes.
- En trait plein, $p''(t)$ la fonction obtenue après filtrage.
- Fait apparaître les évolutions de fond auxquelles le contrôle doit s'appliquer.

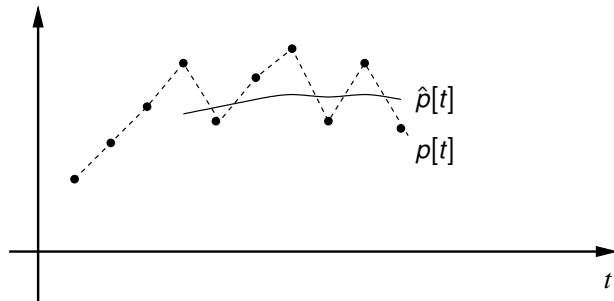
- Toute mesure est entachée d'erreur, donc suivre les variations de trop près peut induire le contrôle en erreur.
- Réduire les conséquences de ces erreurs est l'un des thèmes centraux du domaine dans lequel on parle de *bruit*.
- En réalité, on constate deux formes de bruit :
 - le bruit du processus lui-même qui peut avoir une part d'évolution aléatoire intrinsèque, et
 - le bruit de mesure, lié aux instruments ou procédés d'observation employés.
- Le terme *filtrage* est généralement employé pour désigner l'ensemble des procédés développés pour éliminer le bruit, qu'il provienne du processus ou de la mesure.

- Idée de base : utiliser plusieurs mesures conjointement de manière à réduire les variations dues au bruit.
- Exemple : moyenne des n dernières mesures
 - Soient $p[i], i = 0, 1, \dots$ les mesures bruitées.
 - Produire la séquence filtrée $\hat{p}[i] = \frac{\sum_{j=i-n+1}^i p[j]}{n}, i \geq n$
- C'est ce qu'on appelle un filtre à *moyenne mouvante*.
- Exemple : pour $n = 4$

t	0	1	2	3	4	5	6	7	8	9
$p[t]$	1.0	1.3	1.5	1.8	1.4	1.7	1.9	1.4	1.8	1.3
$\hat{p}[t]$				1.4	1.5	1.6	1.7	1.6	1.7	1.6

Comparatif $p[t]$ versus $\hat{p}[t]$

- Sur les données de l'exemple précédent :

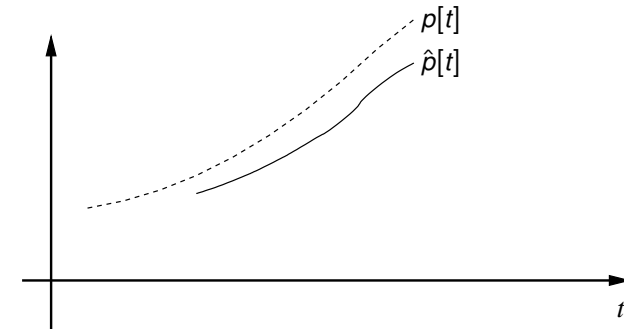


- On constate que $\hat{p}[t]$ lisse bien les mesures bruitées et produit une série de mesures filtrées dégagée des variations aléatoires autour d'une évolution générale fondamentale.

Problème potentiel : inertie du filtrage

- Nouvelles données :

t	0	1	2	3	4	5	6	7	8	9
$p[t]$	1.0	1.2	1.5	1.9	2.4	3.0	3.7	4.5	5.4	6.4
$\hat{p}[t]$				1.4	1.75	2.2	2.75	3.4	4.15	5.0



- Dans cet exemple, les mesures filtrées sous-estiment systématiquement les mesures brutes, mais on pourrait observer toutes sortes d'autres distorsions (déphasages, ...).

Filtres plus performants

- Il existe d'autres (meilleurs) filtres que la moyenne mouvante.
- Filtre exponentiel : avec $0 \leq \alpha \leq 1.0$

$$\hat{p}[j] = \alpha p[j] + (1 - \alpha) \hat{p}[j - 1]$$

- Par α , on règle les poids respectifs des nouvelles valeurs par rapport aux anciennes.
 - $\alpha \rightarrow 1.0$ si on veut suivre et réagir aux évolutions du plus près, et
 - $\alpha \rightarrow 0.0$ si on veut limiter les effets du bruit ponctuel.
- Filtres à *prédiction linéaire* : les valeurs précédentes sont utilisées pour calculer la dérivée (supposée constante) à partir de laquelle on prédit la valeur courante, prédiction qui est ensuite combinée à la mesure courante pour produire une valeur filtrée.
 - Filtre de Kalman : filtre optimal sur prédiction linéaire avec bruit blanc.

Précision et fusion de capteurs

- Fusion de capteurs : combinaison d'informations de plusieurs sources pour produire une information unique plus précise.
 - Complète le filtrage en utilisant l'ensemble des informations disponibles pour estimer le plus précisément possible les valeurs de ses propriétés.
- Ex. : sonar + caméras stéréos pour évaluer des distances.*
- En modélisant les relations (corrélations, etc.) entre les différentes propriétés, la fusion utilise conjointement les mesures pour améliorer la précision les unes des autres ; par exemple :
 - Si p_1 et p_2 sont deux propriétés telles que $p_1(t) = \alpha p_2(t) + \beta$, alors $\hat{p}_2[j-1]$ peut être utilisée pour prédire $p_1[j]$ et $\hat{p}_1[j-1]$ peut être utilisée pour prédire $p_2[j]$.
 - Ces prédictions servent à corriger les mesures, selon des principes similaires à ceux des filtres mais sur plusieurs propriétés.
 - Cependant, toutes les mesures utilisées dans une fusion doivent être *cohérentes* i.e., correspondre à un même état.

Conseils pragmatiques pour augmenter la réutilisabilité II

- ⑤ Exposer plusieurs interfaces capteurs et actionneurs définies autour d'un ensemble d'opérations cohérentes en fonction des types de contrôle que l'entité rend possibles.
- ④ Intégrer la configurabilité des capteurs et actionneurs dès le début de la conception et offrir des interfaces de configuration.
- ⑤ Prévoir au déploiement une campagne de simulation pour l'identification du modèle de comportement, l'estimation des méta-propriétés des capteurs et la caractérisation des actionneurs par des opérations automatisées.
- ⑥ Documenter le plus explicitement possible les méta-données et les méta-propriétés et prévoir les moyens de leur mise à disposition pour les entités de contrôle.

Récapitulons...

- 1 L'entité adaptable est le **parent pauvre** de l'informatique autonome.
- 2 Ses **fonctions capteurs et actionneurs** et son comportement sous adaptation sont pourtant la **pierre angulaire** d'un système auto-adaptable.
- 3 Concevoir une entité adaptable dans un contexte **ouvert** est **difficile**, et ne peut être réalisé sans une **explicitation** complète de ses propriétés.
- 4 Une entité adaptable doit être conçue autour d'un **modèle de comportement**, intégrant ses opérations d'adaptation et ses modes de fonctionnement.
- 5 La **mesure** de propriétés est un processus qui peut être coûteux et qui peut perturber de manière importante l'entité mesurée.
- 6 De plus, sur les systèmes informatiques, les propriétés continues ne peuvent être mesurées que par des **interventions discrètes**, ce qui soulèvent des difficultés d'**échantillonnage**.
- 7 Les **erreurs**, inhérentes la mesure, doivent être appréhendées explicitement, et l'utilisation de **filtres**, malgré leurs inconvénients, est souvent une nécessité.

Pour aller plus loin : sélection de lectures recommandées

- *An architectural blueprint for autonomic computing*, IBM, 4^e édition, 2006.
- *Autonomic Computing*, P. Lalanda, J. McCann et A. Diaconescu, Springer-Verlag, 2013, chapitre 4.
- *Stochastic models, estimation and control volume 1*, P.S. Maybeck, Academic Press, 1979, chapitre 1.