Introduction aux systèmes temps réel

Samia Bouzefrane

Maître de Conférences

CEDRIC –CNAM

samia.bouzefrane@cnam.fr
http://cedric.cnam.fr/~bouzefra

Sommaire

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Définitions

•***Un système temps réel*** est un système (application ou ensemble d'applications) informatique dont le fonctionnement est assujetti à l'évolution dynamique d'un procédé extérieur qui lui est connecté et dont il doit contrôler le comportement.

•La correction d'un système temps réel dépend non seulement de la justesse des calculs mais aussi du temps auquel les résultats sont produits [Stankovic 1988] (contraintes temporelles).

•Un système temps réel n'est pas un système « qui va vite / rapide» mais un système qui satisfait des contraintes temporelles (les contraintes de temps dépendent de l'application et de l'environnement alors que la rapidité dépend de la technologie utilisée, celle du processeur par exemple).

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Définitions

•***Un système embarqué*** (*embedded system* ou *système enfoui*) est un système informatique dans lequel le processeur/calculateur est englobé dans un système plus large et où le logiciel est entièrement dédié à une application donnée.
Ex. sonde spatiale, terminal GSM, carte à puce

•Les ressources utilisées pour mener à bien les calculs sont en nombre limité (contraintes matérielles).

•Intervention humaine directe difficile voire impossible.

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemples de grandeur des contraintes temporelles

-La **milliseconde** pour les systèmes de radar

-La **seconde** pour les systèmes de visualisation humaine

-Quelques **heures** pour le contrôle de production impliquant des réactions chimiques

-**24 heures** pour les prévisions météo.

-Plusieurs **mois** ou **années** pour les systèmes de navigation de sonde spatiale.

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Classification

•**Temps réel dur ou critique (hard real-time)**: le non respect des contraintes temporelles entraîne la faute du système.
Ex.: contrôle de trafic aérien, système de conduite de missile, etc.

•**Temps réel souple (soft real-time)**: le respect des échéances est important mais le non respect des échéances ne peut occasionner de graves conséquences.
Ex.: projection vidéo (décalage entre le son et l'image).
Ex.: un robot qui capte des infos sur des objets défilant sur un convoyeur [Duvallet et al. 1999].

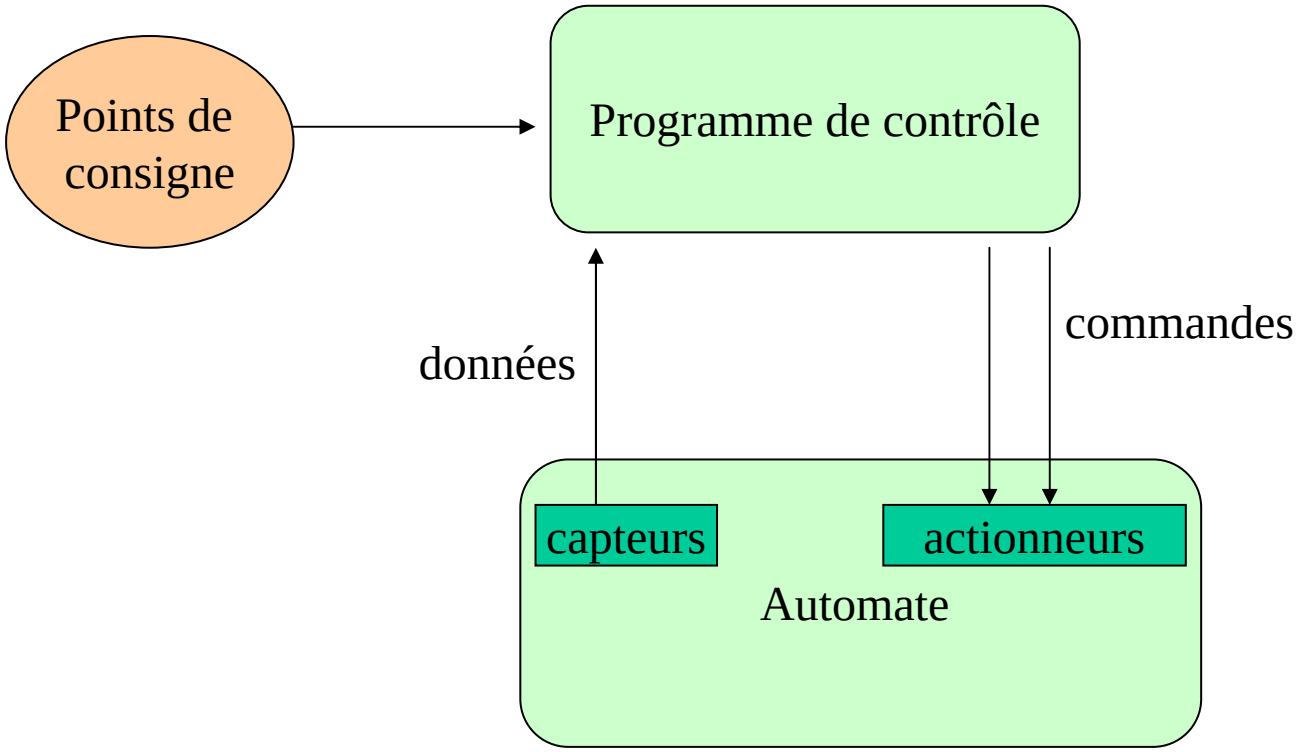
•**Temps réel ferme (firm real-time)**: temps réel souple avec le manquement occasionnel des échéances.
Ex.: projection vidéo (perte de quelques trames d'images).

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Boucle ouverte

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Boucle fermée



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Caractéristiques du temps réel

- **Taille et complexité**
  - -Un système temps réel interagit avec un environnement extérieur souvent complexe et en évolution
  - -Il doit respecter des échéances temporelles, garantir une fiabilité permanente
  - -Il doit pouvoir interagir avec différents types d'éléments matériels.

- **Implémentation efficace**: restrictions dans l'utilisation des constructions du langage

- **Certification**: garantir un fonctionnement conforme aux spécifications

- **Simulation, prototypage**: vérification *a priori* de la conception

- **Plate-formes d'essai**: vérification *a posteriori* du bon fonctionnement

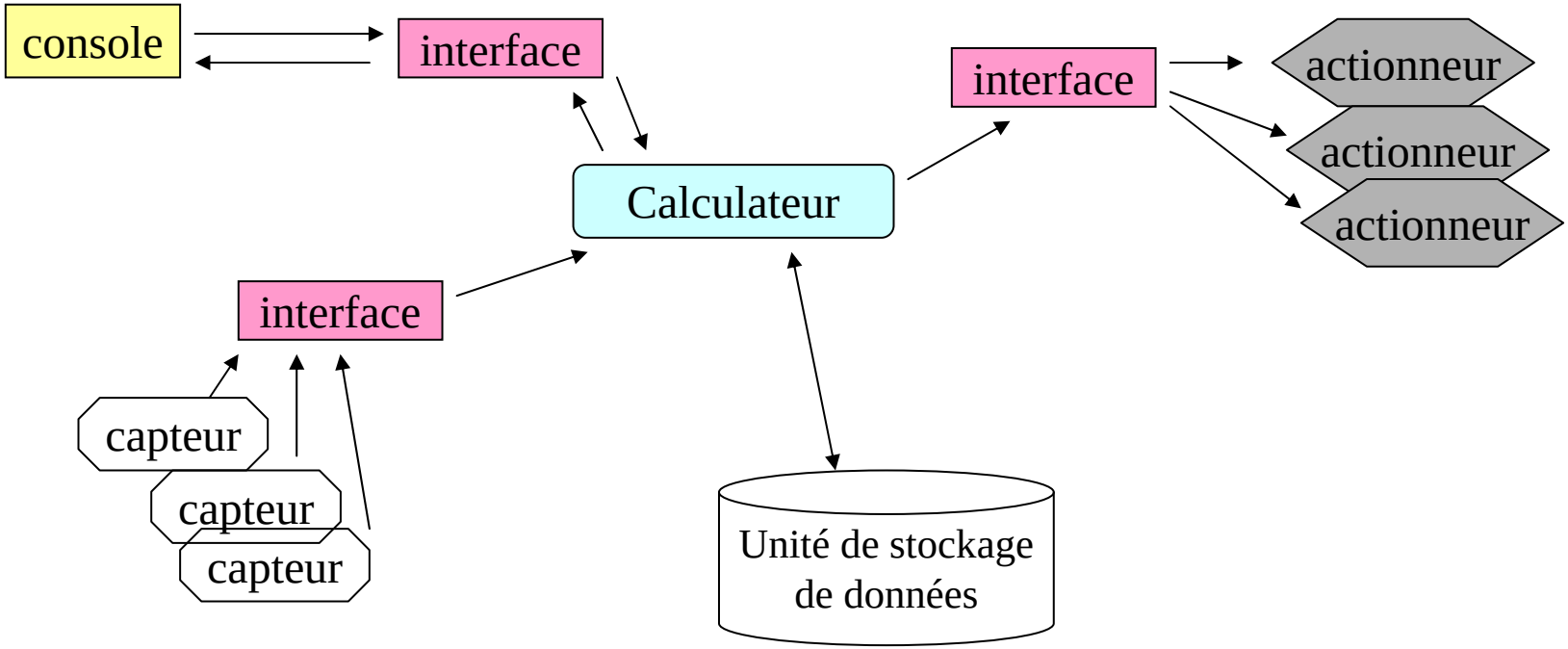
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemples d'applications temps réel

Exemples d'applications temps réel

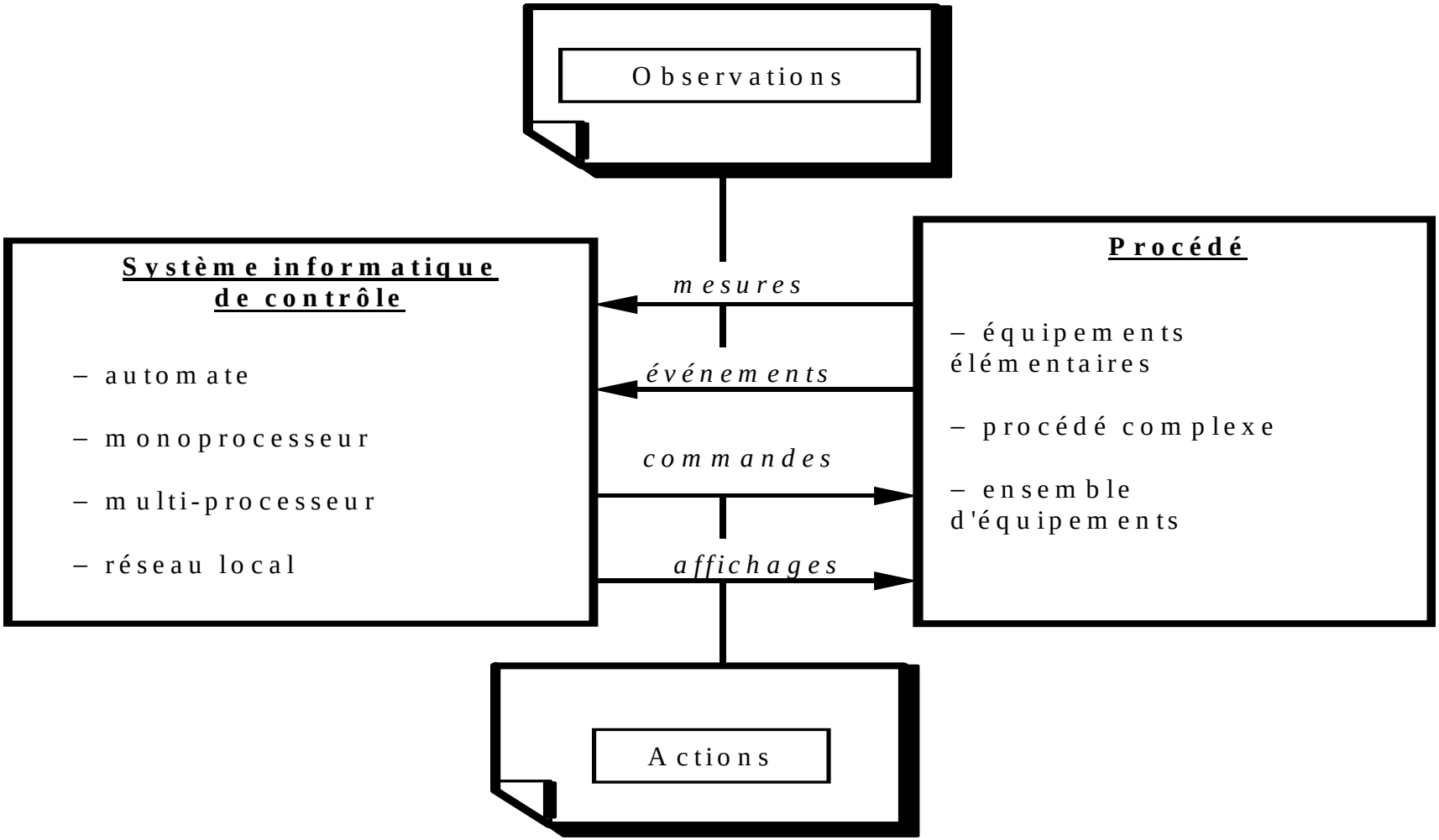
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Un système embarqué/temps réel



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Un système embarqué/temps réel



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemple 1: domaine de l'avionique

- **Système temps réel critique**:

-Contraintes temporelles: échéance, temps de réaction, etc.

-Utilisation de redondance matérielle et logicielle

-Matériel et logiciel dédiés

-Système fermé, validé *a priori*

-Système réparti synchrone: commandes de vol, radars, moteurs, etc.

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemple 1: domaine de l'avionique

•**Dans un Airbus A340** [Boniol 1998]:

Il y a 115 équipements avec :

- 3 calculateurs qui élaborent les paramètres inertiels

- 2 calculateurs qui implémentent les lois de guidage

- 5 calculateurs qui implémentent les lois de pilotage

- 2 calculateurs d'alarmes, etc.

- Environ 200 000 données sont échangées

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemple 2: multimédia sur le Web

- **Système temps réel souple**:

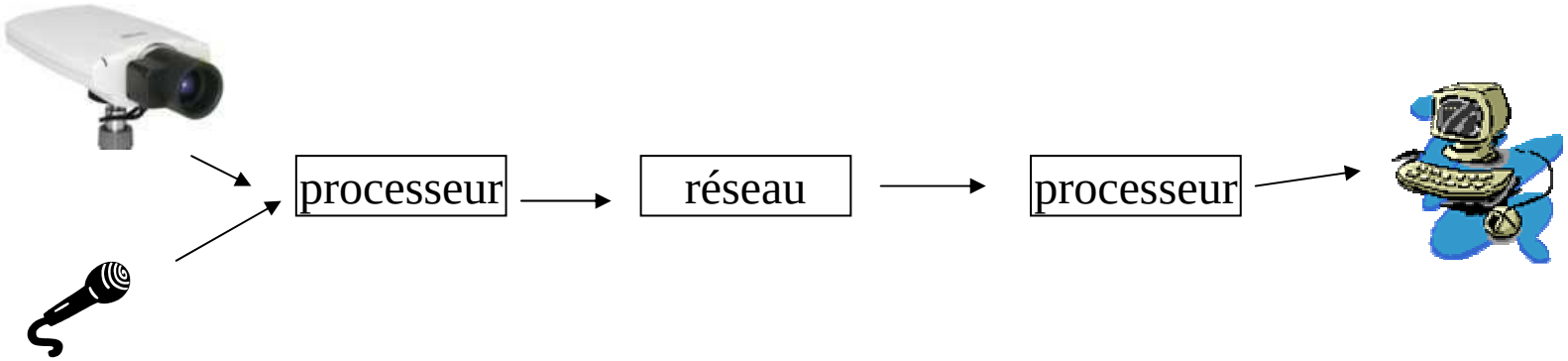
-Contraintes temporelles: gigue, délais de bout en bout, temps de réponse, etc.

-Synchronisations: intra et inter-flux

-Plateforme généraliste (ex. PC + Windows)

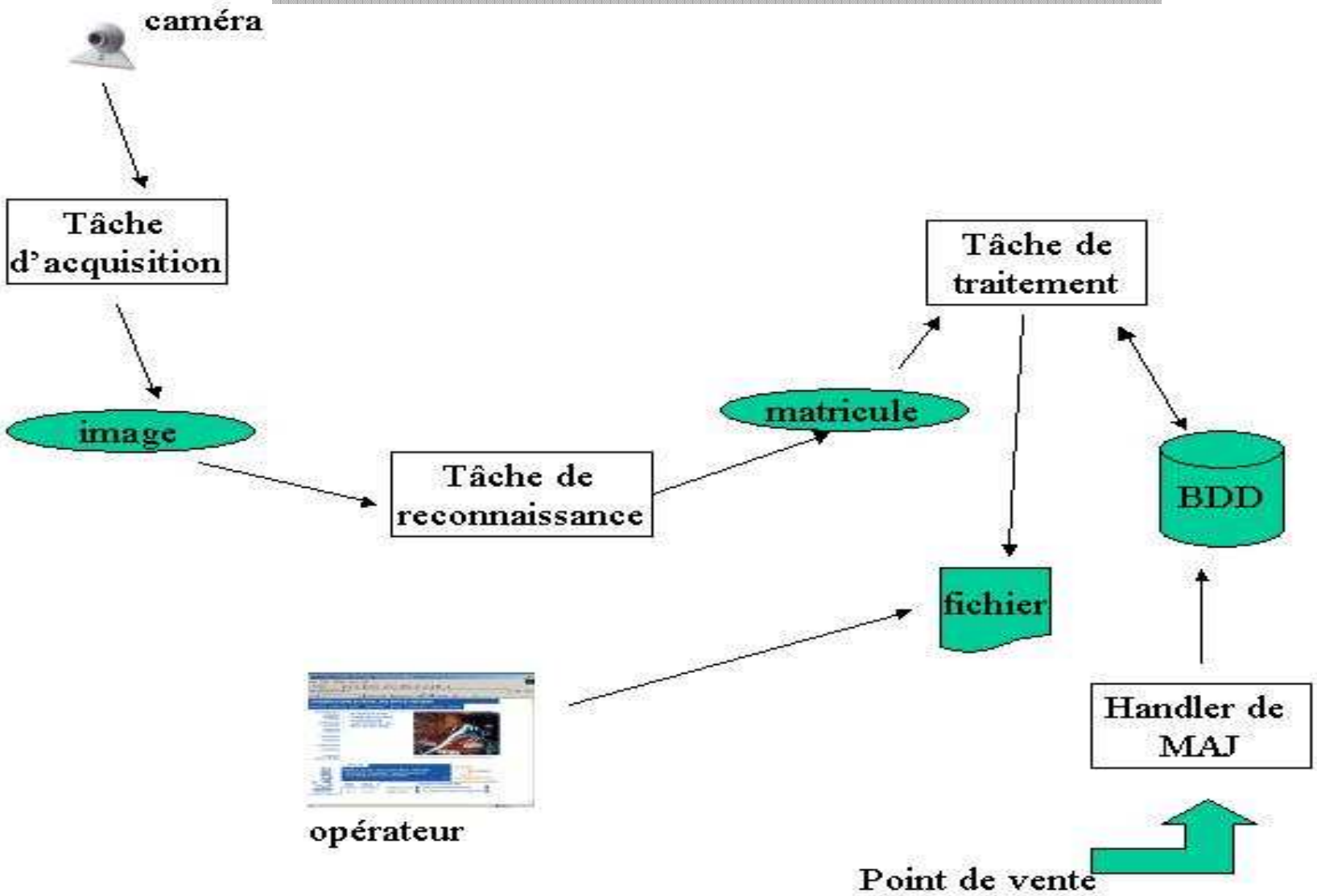
-Application interactive

-Débits variables et difficiles à estimer hors ligne.



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemple 3: Péage au centre de Londres



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Autres exemples d'application

-**Transports** (métro, aérospatiale, automobiles, etc.).

-**Médias** (décodeurs numériques).

-**Services téléphoniques** (terminal GSM, auto-commutateur).

-**Supervision médicale**.

-**Systèmes de production industrielle** : centrale nucléaire, chaîne de montage, usine chimique, etc.

-**Robotique** (ex. PathFinder: sonde lancée par la NASA en mars 1996, composée d'une station au sol et d'un robot mobile *Sojourner*).

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Caractéristiques des applications temps réel

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Utilisation du temps concret

- **Au sein d'une application ou d'un système temps réel, il faut pouvoir manipuler le temps concret (horloge)**

- **Le temps réel (ou temps concret) sera utilisé de plusieurs façons:**
  – Soit en définissant la date à laquelle une action doit être ***commencée***
  – Soit en définissant la date à laquelle une action doit être ***finie***

- **Il peut être nécessaire de pouvoir modifier ces paramètres en cours d'exécution et de pouvoir préciser les actions à prendre en cas de faute temporelle**

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Découpage en tâches

•Dans le monde réel, l'environnement du système temps réel peut consister en plusieurs actions qui évoluent simultanément (en parallèle ou en concurrence).

• Pour réduire la complexité de conception et calquer fidèlement la réalité, il faut utiliser la programmation concurrente :

– utiliser un modèle de tâches ou processus concurrents.

– utiliser des moyens de communication et de synchronisation inter-tâches ou inter-process (mémoire partagée, boîtes aux lettres, files de messages, moniteurs, etc.).

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Respect des contraintes temporelles

- La limitation des ressources (en particulier du processeur) conduit à bloquer des processus (ils ne peuvent progresser du fait du manque de ressource)

- Afin de respecter en permanence les échéances, il faut gérer efficacement la pénurie et tenter de favoriser les processus dont l'avancement est le plus « urgent »

- Un ordonnancement consiste à définir un ordre sur l'utilisation des ressources du système afin de respecter les échéances.

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Ordonnancement

- **On appelle ordonnanceur (scheduler), le processus système qui gère l'ordonnancement des processus (tâches)**

- **Un algorithme d'ordonnancement est une méthode ou stratégie utilisée pour ordonnancer les processus (tâches)**

- **Un tel algorithme s'appuie sur la connaissance de certaines caractéristiques des processus (tâches) ou du système**

– processus (tâches) périodiques ou apériodiques;
– préemption possible ou non;
– échéance et pire temps d'exécution des processus (tâches)
– système à priorité fixe ou à échéance
etc.

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Algorithmes d'ordonnancement

- **Deux algorithmes classiques d'ordonnancement**

– RM (Rate Monotonic) : algorithme à priorité fixe pour tâches périodiques (la tâche la plus prioritaire est celle de plus petite période)

– EDF (Earlest Deadline First) : algorithme à priorité dynamique pour tâches périodiques (la tâche la plus prioritaire est celle de plus petite échéance)

- **L'ordonnanceur choisit d'exécuter la tâche prête de plus haute priorité**

- **Au sein d'une même classe de priorité, le choix peut se faire par temps partagé (Round Robin) ou par ancienneté (gestion FIFO)**

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Cycle de vie d'une application temps réel

Cycle de vie d'une application temps réel

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Génie logiciel et temps réel/1

- **Génie logiciel =** méthodes, modèles et ateliers pour la conception mais aussi pour maîtriser la qualité des produits, leur coût et le respect des délais.

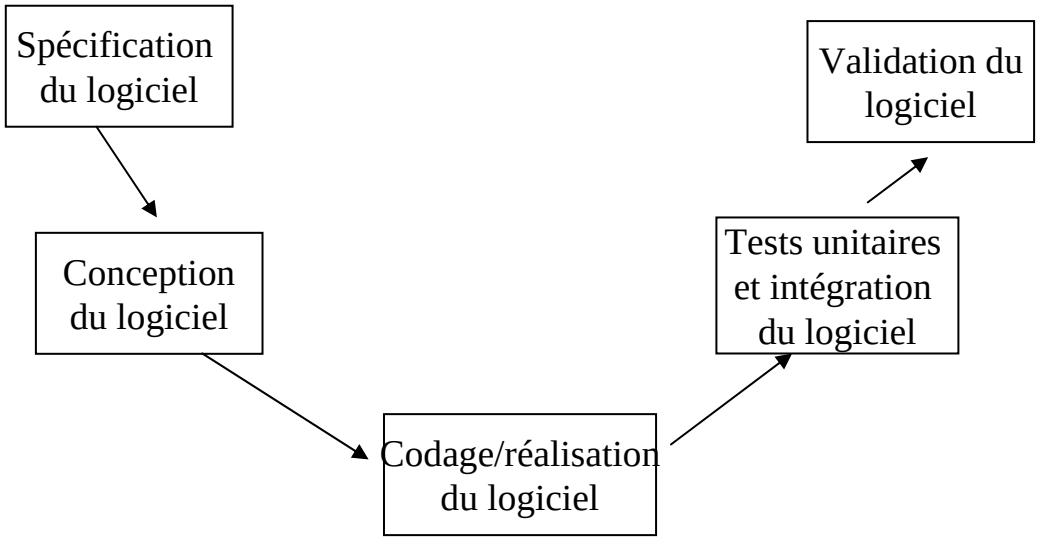
- **Spécificités des applications temps réel**

- Concurrentes et synchronisées
- Manipulation du temps
- Coût de développement très lourd (validation temporelle et logique, applications peu flexibles)
- Maintenance souvent impossible (terminal GSM, sonde spatiale)
- Conséquences tragiques (vies humaines, faillites économiques)

⇒ **Utilisation de méthodes, outils (adaptés aux spécificités du temps réel) qui facilitent la conception et le développement.**

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Génie logiciel et temps réel/2



- Spécification=quoi faire? Conception= comment faire?
- Notions de méthodes, modèles et outils
- Couverture partielle ou totale du cycle. Cela dépend du domaine applicatif

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Méthodes de développement/1

- **Méthodes fonctionnelles structurées**
  - SA_RT : Structured Analysis Real Time [Pirbhai-Hatley 1986]
  - DARTS : Design Approach for Real-Time Systems [Gomaa 1984)]
  - SDL : Specification and Description Language [CCITT 1988]

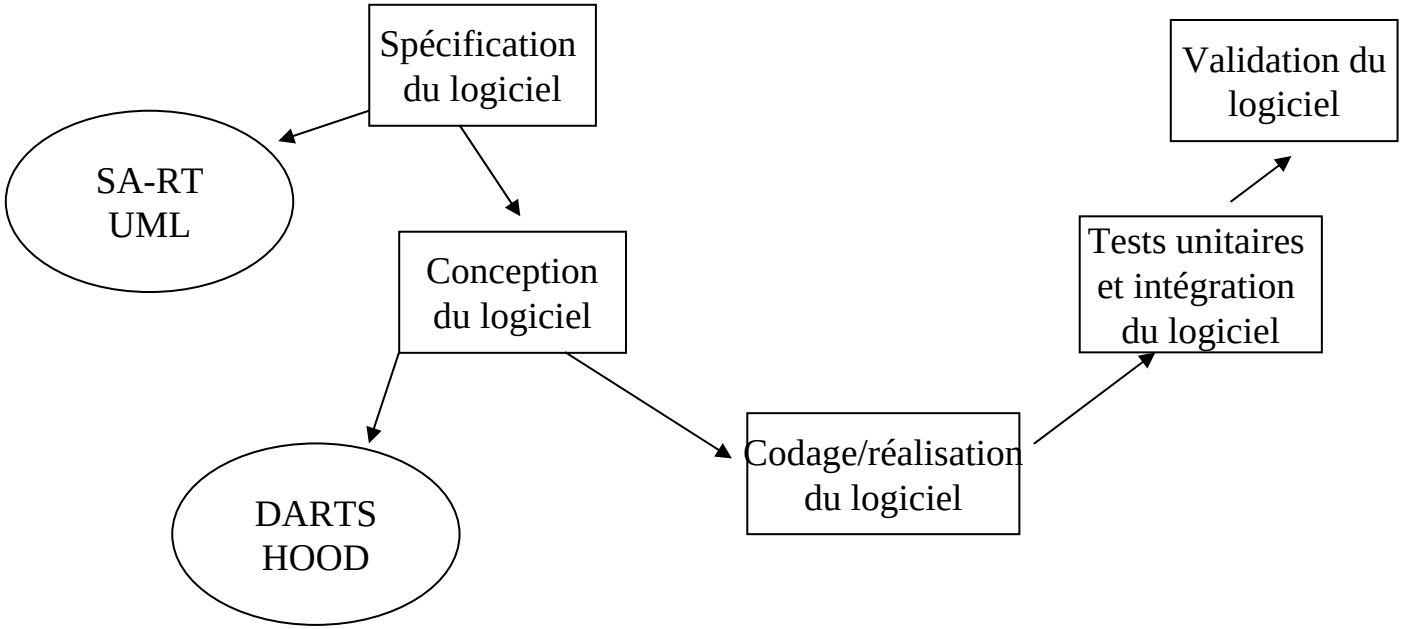
  etc.

- **Méthodes orientées objet**
  - UML : Unified Modeling Language [OMG 1995]
  - HOOD : Hierarchical Object Oriented Design [CRI-Cisi 1987]

- **Méthodes orientées composant**
  - KOALA : technologie composant développée par Philips pour la conception de composants électroniques grand public [Ommering et al. 2002]

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Méthodes de développement/2



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemples industriels

- **Programme Spot4** (Matra Marconi Space/CNES)
  - satellite destiné à l'observation de la terre (météorologie, environnement, etc.)
  - Spécification et conception: HOOD
  - langages : Ada, Assembleur

- **Programme SENIT8** (Dassault Électronique /DCN-Ingénierie)
  - équipements de gestion et contrôle commande du porte-avions Charles de Gaulle
  - Spécification et conception: SART et Ada-Buhr (proche de DARTS)
  - langages : Ada, C

- **Programme Rafale** (Dassault Électronique): avion militaire
  - Spécification et conception: SART et OMT
  - langages : Ada

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Méthodes de spécification et de conception

Méthodes de spécification et de conception

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Spécifier: la méthode SART

- **SART =** Structured Analysis Real Time
  spécification du logiciel

- **Une spécification est constituée de :**

- Dictionnaire de données
- Graphes de flots de données étendus par la notion d'événements
- Diagrammes état-transition pour l'aspect dynamique du système

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

La méthode SART : récapitulation

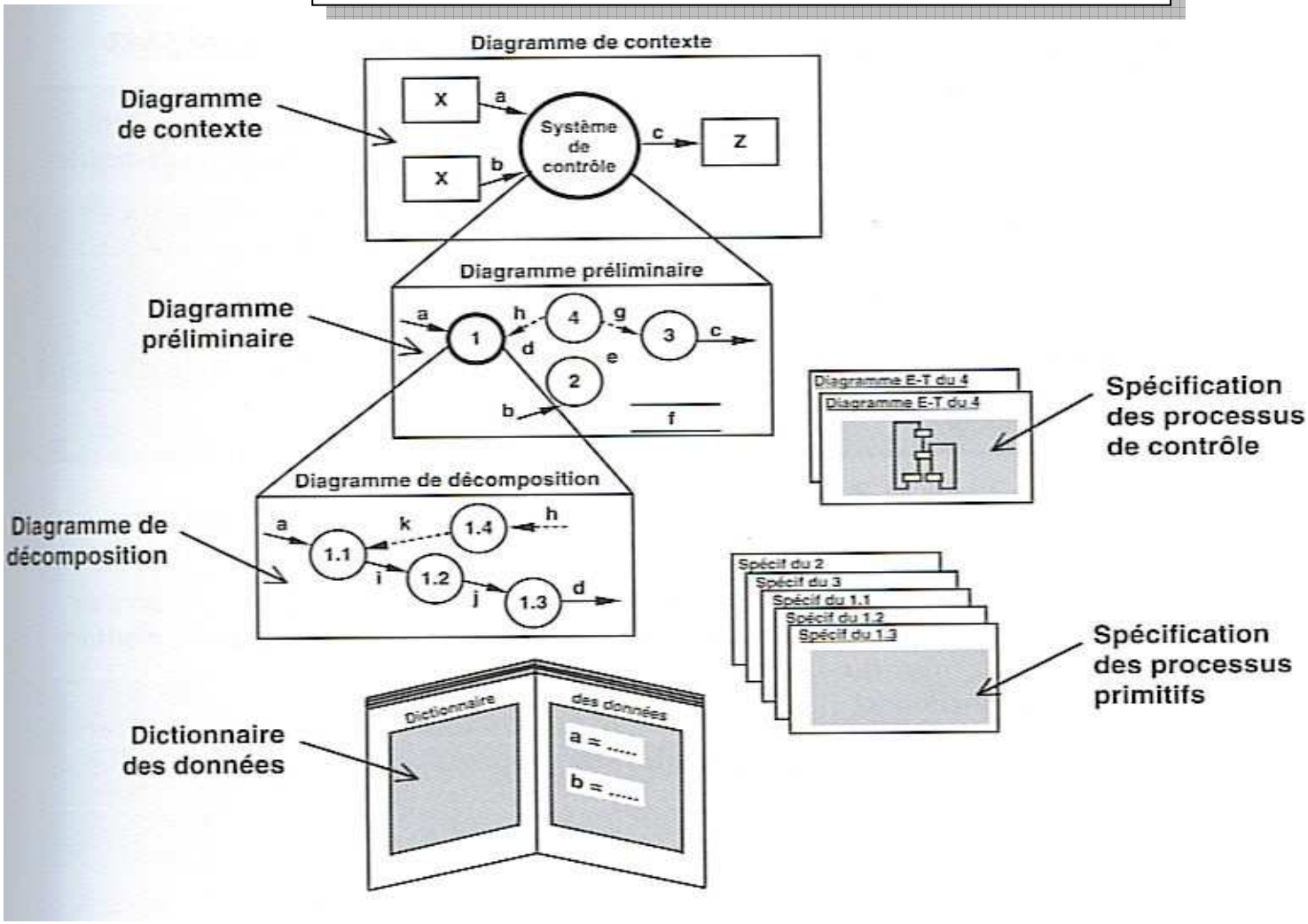


Figure p.55 de [Cottet & Grolleau 2005]

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Concevoir: la méthode HOOD

HOOD = Hierarchical Object Oriented Design (projet ESA (européen) vise à définir une méthode de conception pour des applications spatiales)

- spécification préliminaire et détaillée du logiciel

- 1987: CRI,CISI Ingéniérie, MATRA => HOOD 2.0 Reference Manual

- 1989 : HOOD 3.0 ref. manual

- Exemple : STOOD de la société TNI

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

HOOD en quelques mots

- **Objet: unité de modularité**
  - encapsule données et opérations
  - encapsule aucun, un ou plusieurs flots de contrôle.
  - fournit une interface d'utilisation à ses utilisateurs

- **Objets passifs**
  - fournissent des opérations séquentielles, le flot de contrôle de l'utilisateur est transféré à l'opération demandée

- **Objets actifs**
  - ont un état interne propre
  - ont leur propre comportement
  - fournissent à leurs utilisateurs des opérations "contraintes"

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Concevoir: UML temps réel

- **UML =** Unified Model Language
- Profile temps réel [OMG 2000]

  - stéréotypes pour la gestion du temps

  - stéréotypes pour l'analyse d'ordonnançabilité et la gestion des ressources

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Concevoir: Approche composant

- **Modèle composant temps réel**
  - composant, interface offerte, interface requise, opérations, contraintes de temps, etc.
  - Types de composants: actif, passif, connecteur, composite
  - Événements extérieurs/communication

- **Exemples** :
  - KOALA : technologie composant développée par Philips pour la conception de composants électroniques grand public [Ommering et al. 2002]

  - SAVEComp est une technologie basée sur le modèle SAVECCM destiné au développement du logiciel pour automobiles [Hansson et al. 2004].
  - TinyOS: système non temps réel mais embarqué dans les réseaux de capteur, utilise le modèle composant.

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

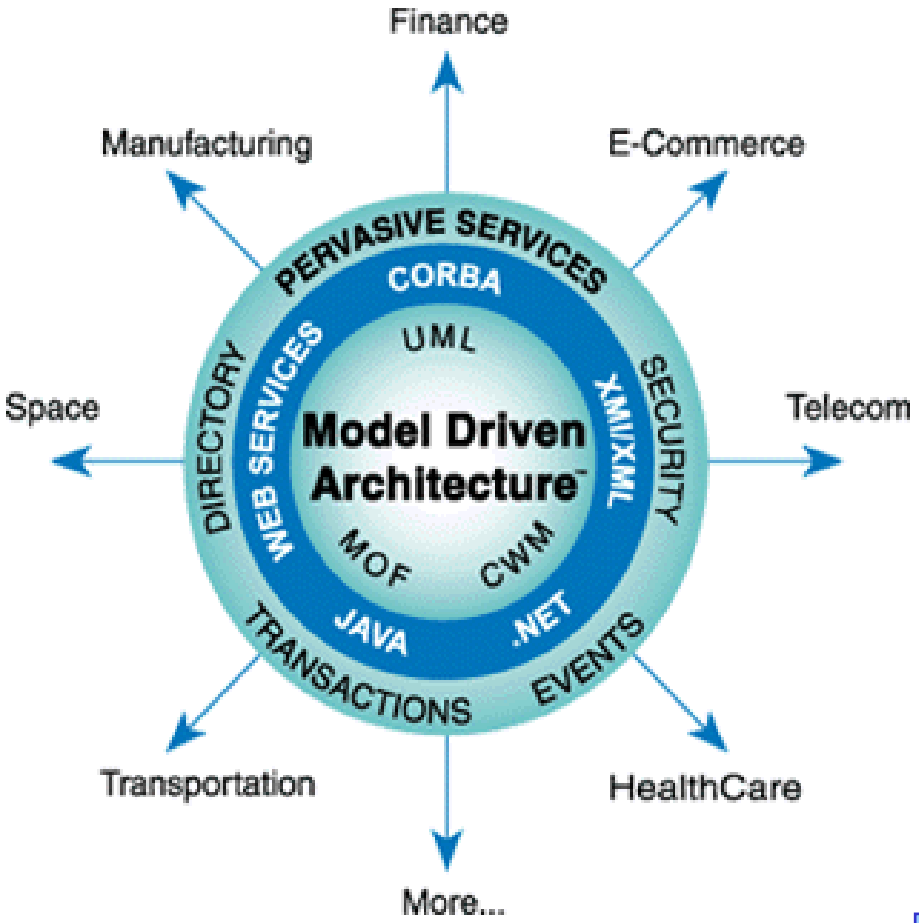
Quelques ADLs

Nom	Focaliser vers	Origine
Wright	Modélisation et analyse du comportement dynamique des systèmes concurrents	CMU
Darwin	Systèmes massivement distribués	ICL
Rapide	Modélisation et simulations du comportement dynamique des architectures	Stanford
UniCon	Génération de code de liaison pour l'interconnexion de composants préexistants	CMU
MetaH	Conception, validation et génération d'applications temps réel embarquées (avionique)	Honeywell

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Le Model-Driven Architecture (MDA)

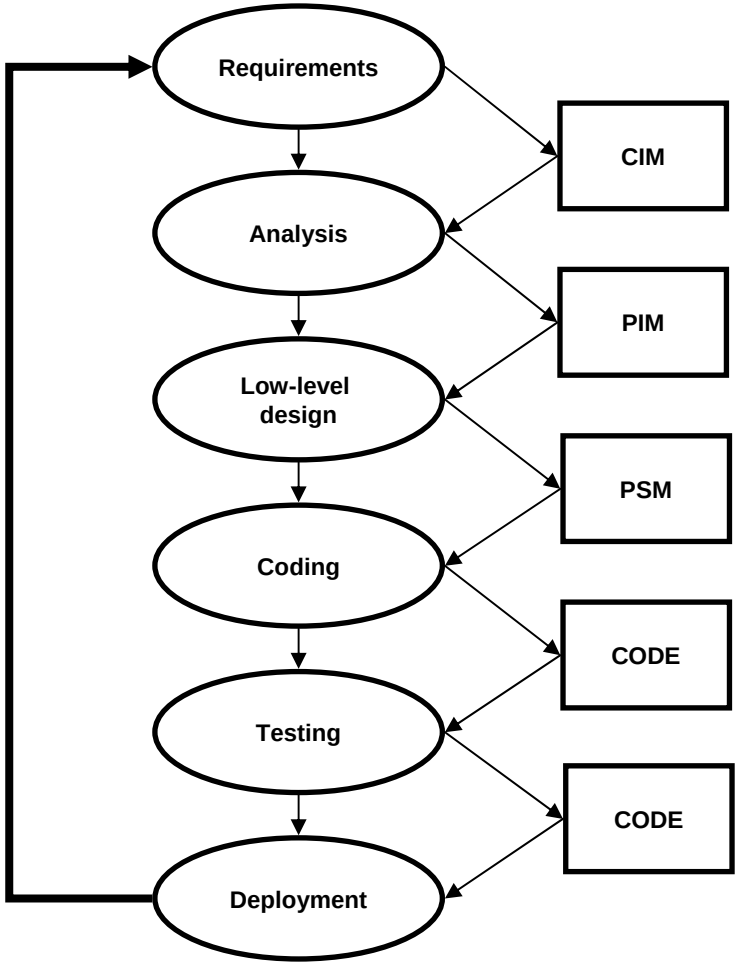
- **Proposé par l'«Object Management Group» (OMG)**
  - Permet la séparation des spécifications fonctionnelles du système des spécifications concernant son implantation sur plate-forme technologique quelconque
  - Le principe du MDA tourne autour de modèles et de transformation de modèles.

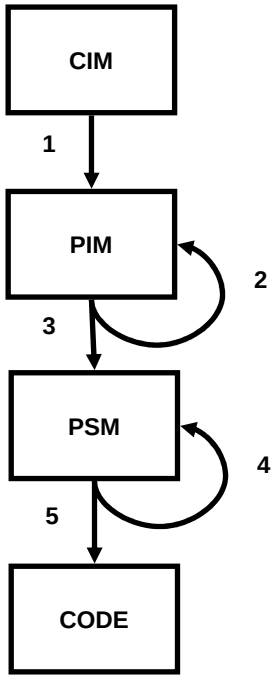


samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Le processus MDA

- **Supporte toutes les étapes du cycle de développement logiciel**
  - Des modèles sont créés à chaque phase de développement
  - Les passages entre les différentes phases se font à l'aide de projections standardisées et automatiques.





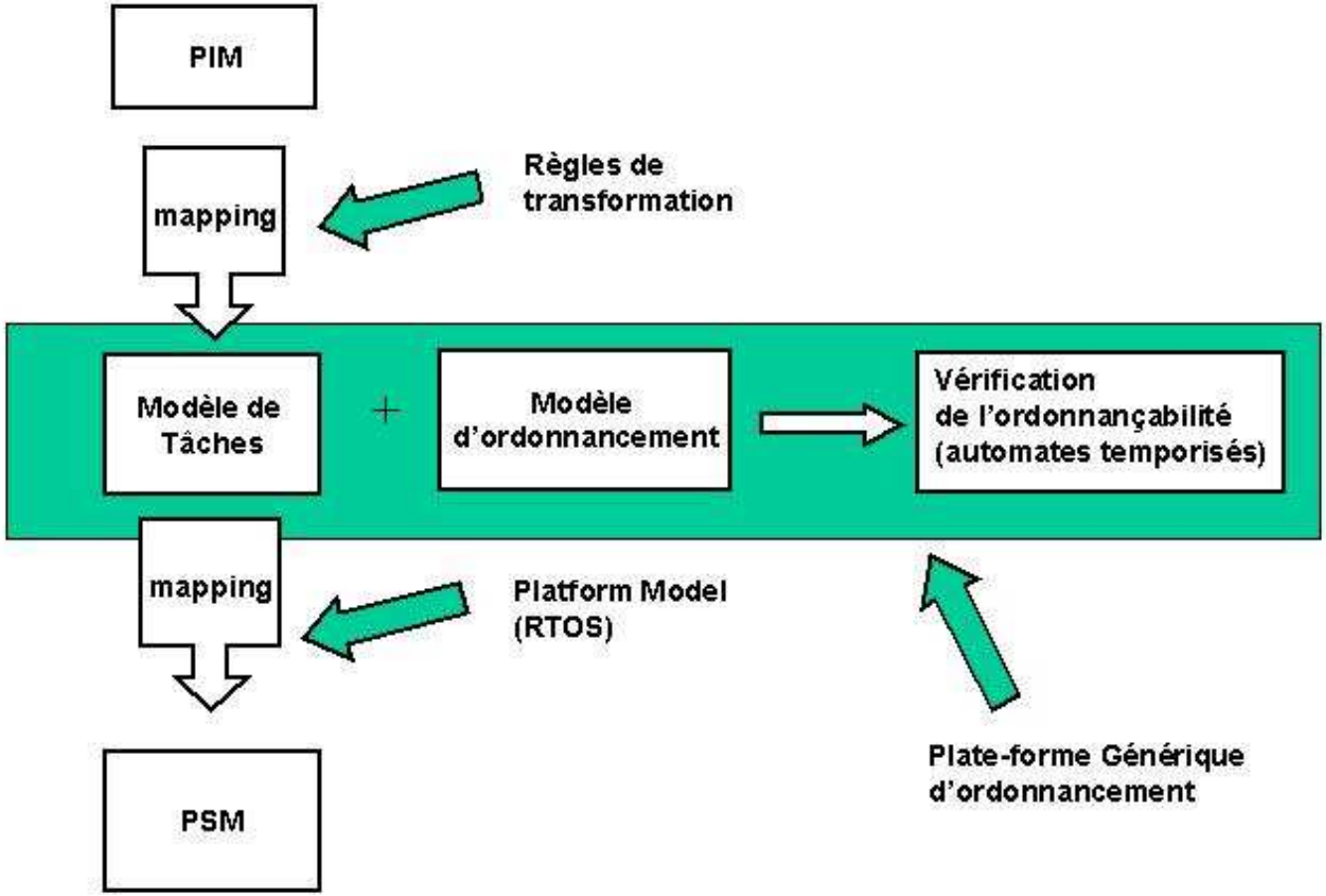
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les transformations de modèles

- **PIM vers PIM**
  - Vise à enrichir, filtrer ou spécialiser le modèle sans nécessiter d'informations dépendantes d'une plate-forme
- **PIM vers PSM**
  - Utilisé quand le PIM est suffisamment raffiné pour être projeté vers une plate-forme d'exécution
- **PSM vers PSM**
  - Utilisé lors des phases de déploiement, d'optimisation, de reconfiguration ou lors de la génération du code
- **PSM vers PIM**
  - Utilisé pour obtenir un modèle d'abstraction PIM à partir d'une implantation existante sur une plate-forme spécifique (reverse engineering)
    - Difficilement réalisable

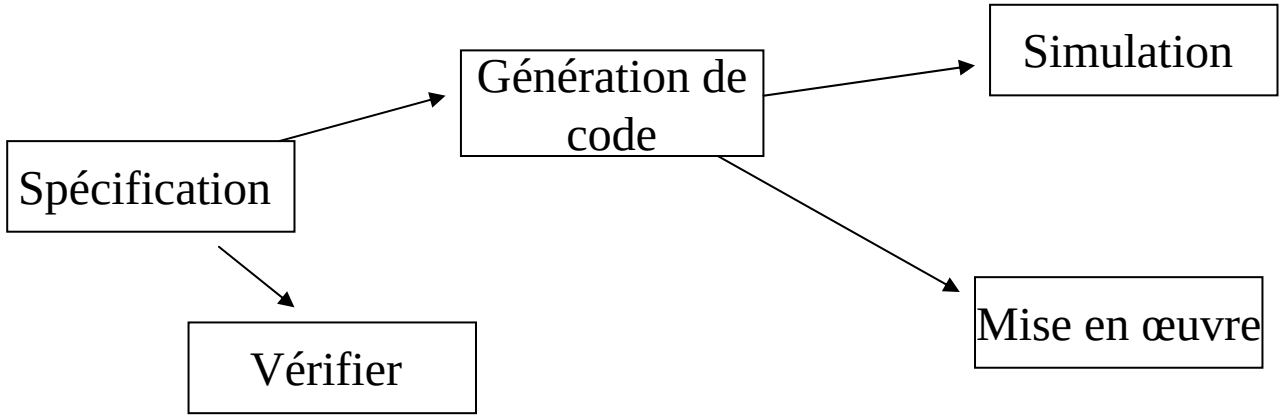
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Concevoir: Approche MDA



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les approches formelles

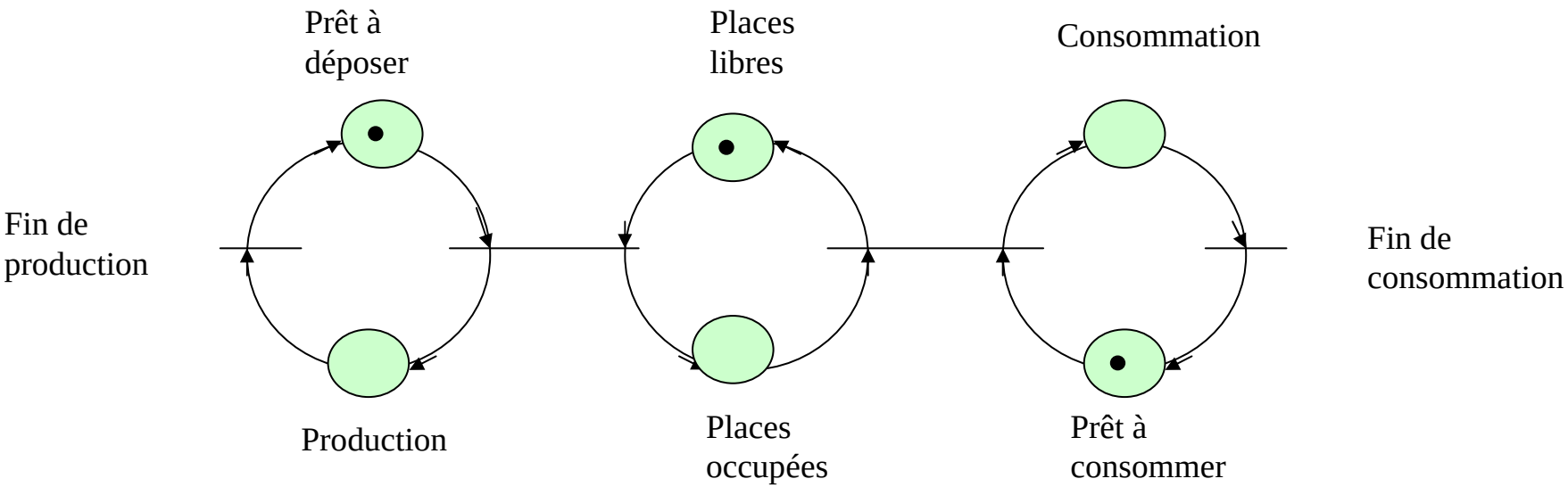


Objectifs

- Spécifier et/ou concevoir formellement un système
- Valider des propriétés (ex. temporelles)
- Génération de code pour effectuer des simulations
- Dans le futur, être capable d'embarquer le logiciel

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les réseaux de Pétri



- Place, jeton, transition. Modèle à états
- Validation logique, éventuellement temporelle
- Étude du graphe d'états: vivacité (interblocage), respect d'invariant (ex. exclusion mutuelle)

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Langages pour le temps réel

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Famille de langages possibles

- **Trois sortes de langages peuvent être identifiés dans le contexte du développement d'applications temps réel**

– les langages assembleurs

– les langages séquentiels liés à des librairies système

– les langages concurrents de haut niveau

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les langages de type assembleur

- **Historiquement, ces langages furent longtemps les seuls à être utilisés dans ce contexte**

- **Dépendant par nature de l'architecture cible (matériel et système)**

- **Aucune abstraction possible et grande difficulté de développement, de maintenance et d'évolution**

-> Langages à proscrire sauf pour l'implémentation de petites fonctionnalités très spécifiques et apportant une grande amélioration des performances

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les langages séquentiels

- **Introduits pour remédier aux problèmes dus au codage en assembleur**

- **Les plus connus sont le C, le C++ ou encore le Fortran**

- **Apporte un plus grand pouvoir d'abstraction et une certaine indépendance du matériel**

- **Mais, doit faire appel à des librairies systèmes spécifiques pour la manipulation des processus**

-> Ces langages posent le problème de la standardisation des appels systèmes mais sont quelques fois le seul choix possible à cause de la spécificité d'une cible et des outils de développement sur celle-ci

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les langages de haut niveau

- **Langages généralistes incluant de plus la notion de tâches et des primitives de synchronisation**

- **Haut pouvoir d'abstraction, indépendance des architectures et des systèmes cibles (ou très peu dépendants)**

- **Parmi ces langages, Ada et Java temps réel sont des langages qui peuvent être utilisés à profit dans le développement d'applications/systèmes temps réel**

-> Langages à privilégier lorsque d'autres contraintes (manque de formation, reprise de code existant, coopération inter-équipes/ ou inter entreprises, ...) ne rendent pas la chose impossible

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Langages choisis dans ce cours

- Java comme exemple de langage concurrent de haut niveau
- C avec l'utilisation de bibliothèques système
- Java temps réel (jRate une implémentation de RTSJ :Real-Time Specification for Java, sous Linux)

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Le choix d’un exécutif temps réel

Le choix d’un exécutif temps réel

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les systèmes d'exploitation

- **Caractéristiques d'un système d'exploitation**

– approche généraliste

– supporte généralement plusieurs types d'applications simultanément

– interaction par appels système

– peu dépendant du domaine d'applications visé

– généralement de taille plus importante qu'un exécutif

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exécutif temps réel/1

- **Caractéristiques d'un exécutif temps réel**

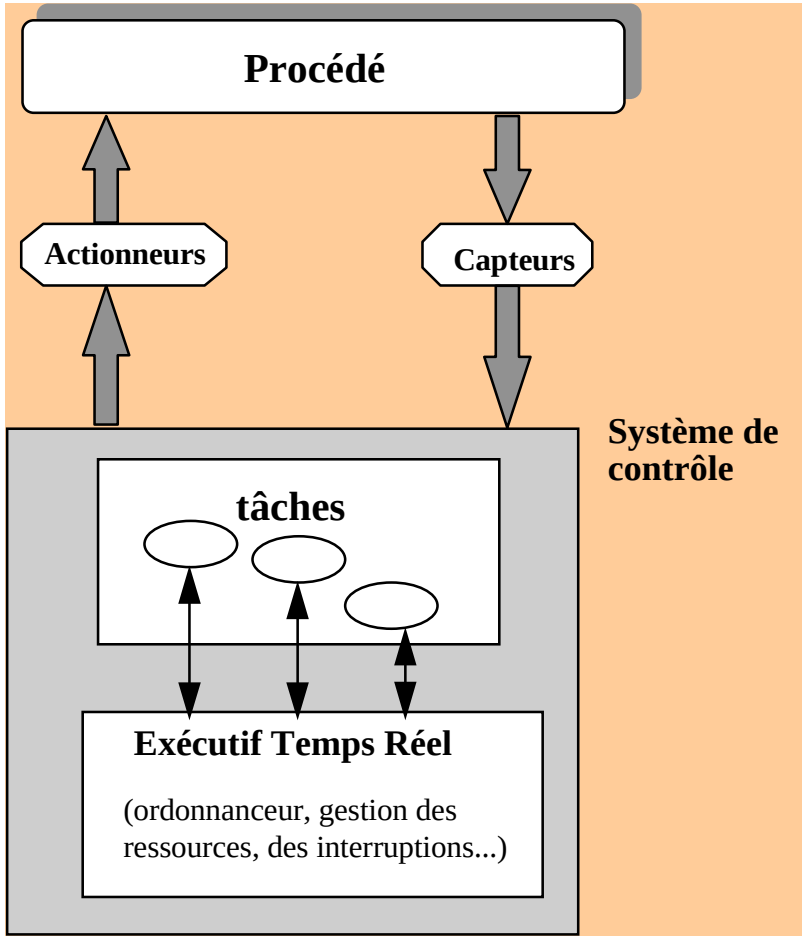
– système spécialisé

– dédié à une application spécifique (ex. contrôle de trafic aérien)

– collection de primitives

-> plus spécialisé qu'un système d'exploitation, code de taille plus petite qu'un système classique

Exécutif temps réel/2



Structure d'un noyau temps réel

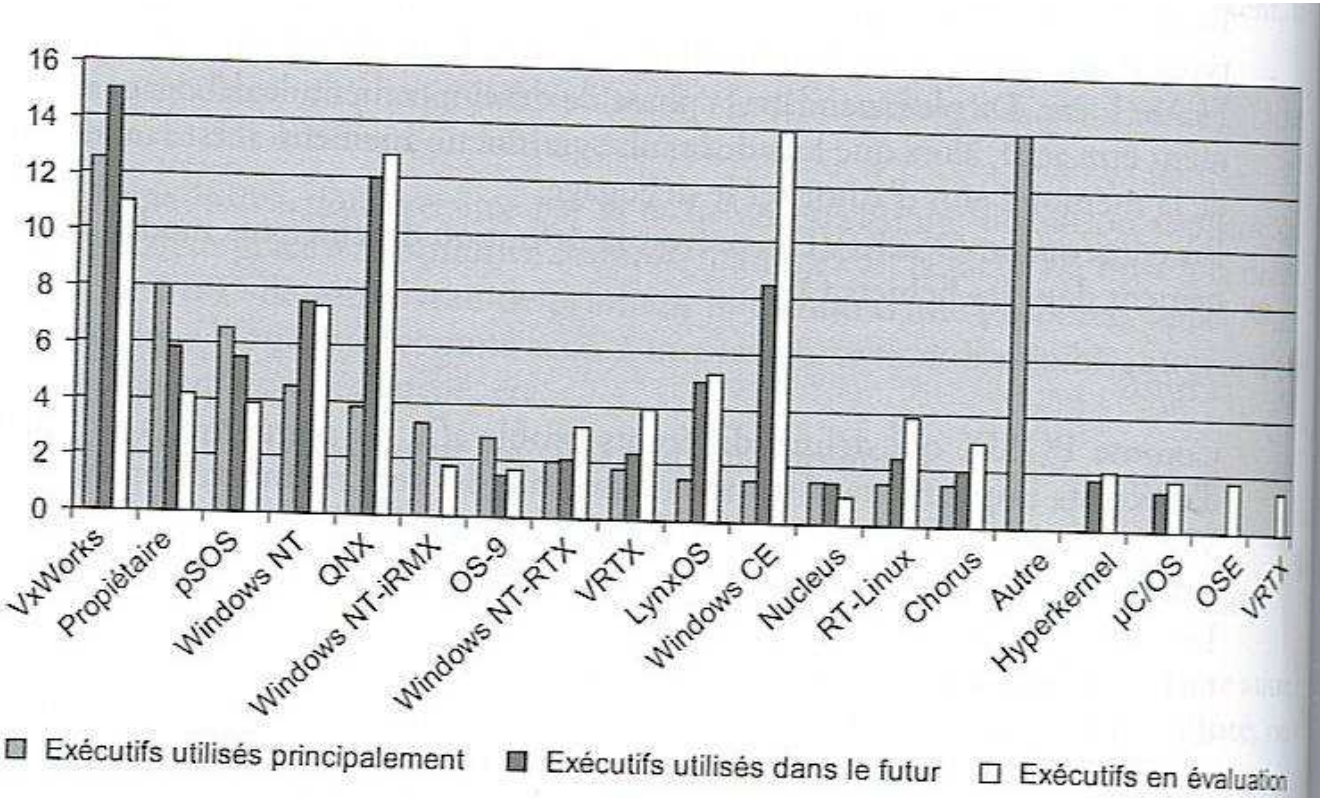
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemples d'exécutifs temps réel

- RTAI: basé sur du code libre, extension de Linux (http://www.aero.polimi.it/~rtai/index.html)
- Lynx-OS : système Unix à base de thread noyau (http://www.lynuxworks.com/) compatible avec Linux
- QNX : système Unix (http://www.qnx.com/)
- Windows CE : système Microsoft temps réel (http://www.cewindows.net)
- VxWorks : Exécutif de Wind river (http://www.windriver.com)
- VRTX : (http://www.mentor.com/vrtxos/)
- RI/RTSJ (Machine virtuelle): Java TR de TimeSys (http://www.timesys.com/company/java.htm)
- jRate/RTSJ: http://linux.softpedia.com/get/Programming/Compilers/jRate-16990.shtml

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les exécutifs temps réel les plus utilisés



Sondage effectué par Open Group en 2001 auprès de 10 000 utilisateurs

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Conclusion

- **Les systèmes ou applications temps réel sont**

– complexes

– font intimement intervenir le temps dans leur conception

– ont des besoins de fiabilité importants

– généralement décomposés en sous-systèmes avec des tâches ou des processus qui interagissent

– doivent être implémentés avec des langages appropriés

– doivent être exécutés sur des systèmes ou des exécutifs adaptés

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Références

[CRI-Cisi 1987]: HOOD Manual, CRI-Cisi Matra, Toulouse.
[Boniol 1998]: Frédéric Bonio, « Une approche synchrone multi-formalismes pour la conception de systèmes temps-réel distribués », TSI 1998.
[Cottet & Grolleau 2005]: F. Cottet & E. Grolleau, « systèmes temps réel de contrôle-commande : conception et implémnetation », Ed. Dunod, 2005, ISBN:2 10 007893 3.
[Dorseuil & Pillot 1991] : Dorseuil A. et Pillot P., « le temps réel en milieu industriel », édition Dunod, collection informatique industrielle, 1991.
[Duvallet et al. 1999]: C. Duvallet, Z. Mammeri et B. Sadeg, « Analyse des protocoles de contrôle de concurrence et des propriétés ACID dans les SGBD temps réel », Revue TSI, 1999.
[Gomaa 1984)]: Gomaa, Hassan, “A Software Design Method for Real-Time Systems,” Communications of the ACM, Sept., 1984
[Hansson et al. 2004]: H. Hansson, M. Akerholm, I. Crnkovic & M. Törngren, "SaveCCM: a component model for safety-critical real-time systems", EuroMicro Conference, Special Session Component Models for Dependable Systems, Rennes, France, Sept. 2004.
[Jaulent 1994]: P. Jaulent, « Génie Logiciel: les méthodes », Armand Colin, 1994.
Jean-François Peyre, supports de cours sur l’informatique industrielle-systèmes temps réel, CNAM(Paris).
Frank Singhoff, supports de cours sur le temps réel, département informatique, université de Bretagne Occidentale (http://beru.univ-brest.fr/~singhoff/supports.html).
[Ommering et al. 2000]: R. Van Ommering, F. Van der Linden & J. Kramer, "*The Koala component model for consumer electronics software*", IEEE Computer, Vol. 33, N° 3, pp. 78-85, March 2000.
[Pirbhai-Hatley 1987]: D. J. Hatley, I. A. Pirbhai: Strategies for Real-Time System Specification; Dorset House, New York, 1987.
RTSJ: http://www.rtsj.org/
[Stankovic 1988]: John Stankovic, « Misconceptions about real-time computing », IEEE Computer, oct. 1988.

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -