Département Informatique INSA de Lyon

Conception et Intégration d'Architectures Industrielles / Systèmes d'Exploitation Avancés

Jean-Philippe Babau

Département Informatique, INSA Lyon

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

jean-philippe.babau@insa-lyon.fr

Plan des enseignements SEA et CIAI

- Domaine
- Spécification (CIAI)
  - SA-RT
    - 1 TD(JPB), 1 projet (JPB/MMi/LM)
  - contraintes temporelles
- Architecture matérielle et logicielle (SEA)
  - architecture matérielle
  - les exécutifs temps réel
    - 2 TDs (RA)
  - ordonnancement
  - carte à puce

jean-philippe.babau@insa-lyon.fr

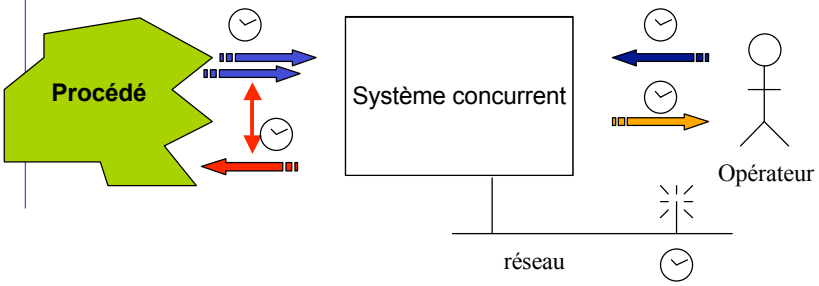
Plan des enseignements SEA et CIAI

- Conception multitâches (CIAI)
  - LACATRE / VxWorks
    - 1 projet (SG/BR)
- Pilote intégré de périphériques (SEA)
  - pilote sous VxWorks
    - 1 TD(JS), 1 TP (LM, JPB, JS)
    - 2 TD (RA)
- Communication (CIAI)
  - réseaux dédiés
  - réseaux de terrain (CAN)
    - 1 TD (JPB)

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Les systèmes temps réel embarqués communicants

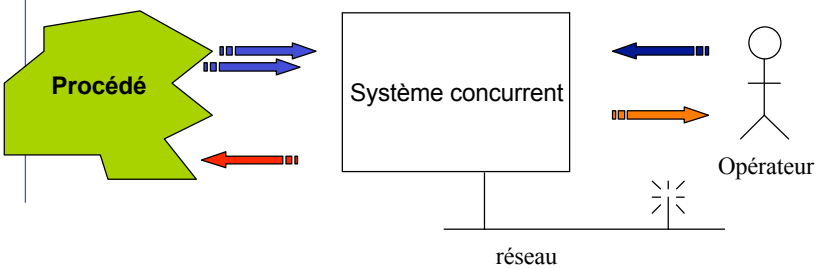


Contraintes temporelles
- acquisition, sorties, échéances

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Les systèmes temps réel embarqués communicants



Contraintes temporelles
- acquisition, sorties, échéances

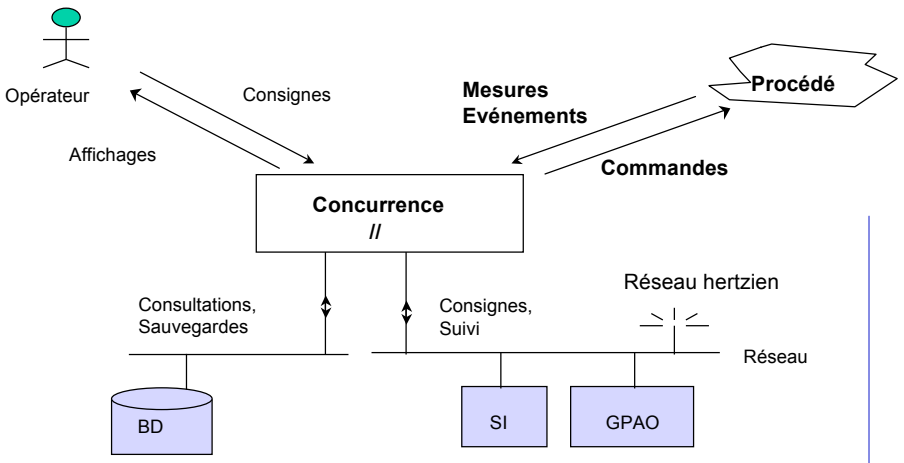
Contraintes d'embarquabilité
- Place limitée, contraintes physiques
- Contraintes d'énergie

Contraintes de coût

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Système réactif



jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Domaines d'application

- Périphériques
  - imprimantes, modems, claviers, souris
- Domotique
  - HiFi, électroménager, confort
  - contrôle à distance, régulation
- Systèmes automatisés
  - contrôleurs industriels
  - système de mesure
- Systèmes mobiles
  - carte à puce
  - téléphones mobiles, PDA
  - véhicule : automobile, avion, train, fusée, robot
- Systèmes à réalité virtuelle
  - simulateurs de conduite, télé-opération

jean-philippe.babau@insa-lyon.fr

Evolutions actuelles du domaine

- Transport
  - Automobile
    - Plus de 50 ECU (Electonic Control Unit)
  - Avionique
    - Airbus A300 (1974) : 25 ko
    - Airbus A380 (2005) : 64 Mo
  - Spatial
    - Spot1 (1980) : 48 ko
    - Mars Express (2003) : 1,2 Mo
- Systèmes personnels
  - Systèmes portables communicants
    - Pda, lecteurs mp3, smartphone
  - Image
    - Appareil photo numériques, …
- Capteurs
  - Médical, industriel, identifiants, suivi

jean-philippe.babau@insa-lyon.fr

Définitions

- Un système est dit concurrent lorsqu'il est composé de modules pouvant s'exécuter en parallèle
- Un système temps réel possède des contraintes temporelles explicites
- Un système est dit embarqué lorsqu'il est physiquement lié au procédé à contrôler
- Un système est dit enfoui lorsqu'il est caché en utilisation
- Un système est dit critique lorsqu'une défaillance du système à des conséquences grave
  - environnement
  - procédé à contrôler
  - personnes

jean-philippe.babau@insa-lyon.fr

Autres définitions

Real-time system

"A real-time computer system may be defined as one which controls an environment by receiving data, processing them, and returning the results sufficiently quickly to affect the environment at the time"

"Pertaining to processing of data by a computer in connection with another process outside the computer according to time requirements imposed by the outside process"

Embedded system

"Anything that uses a computer but does not look like one"

"The microprocessor in an embedded system is like an electric motor in a washing machine"

"Software to control the universe"

"An embedded system means the real-time software is a component of a larger HW/SW system"

jean-philippe.babau@insa-lyon.fr

Temps réel

- Arriver à l'heure ≠ aller vite
- Temps réel dur
  - une échéance stricte est à respecter

  «un résultat juste mais hors-délai est un résultat faux»
- Temps réel mou ou relâché
  - tolérance de dépassement d'échéance
    - pourcentage de non respect
    - taux de dépassement
- Quelques unités de temps
  - milliseconde : systèmes radar
  - seconde : système de visualisation
  - minute : chaîne de fabrication
  - heure : contrôle de réaction chimique

jean-philippe.babau@insa-lyon.fr

Définition de l'heure

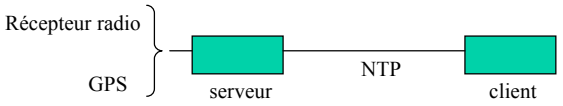
- Jour solaire
  - 2 passages successifs au zénith
  - 1 jour : 24h de 60 min de 60s ()
- Seconde
  - 1 / 86 400 ème de jour
  - 9 192 631 770 périodes de la radiation correspondant à la transition entre deux niveaux hyperfins de l'état fondamental de l'atome de Césium 133
- TAI (Temps Atomique International)
  - nombre moyen (4 horloges) de tops horloge atomique à base de césium 133 depuis le 1er janvier 1958 divisé par 9 192 631 770 (1 seconde)
  - Échelle de temps linéaire et stable : mesuré par 250 horloges atomiques
- UTC (temps universel coordonné)
  - Heure officielle
  - Ajustement avec le TAI si écart (TAI – temps solaire moyen) > 900 ms
    - 1 seconde est sautée
    - En moyenne une seconde par an
    - 31 décembre 2005 : 1 seconde de plus
    - UTC – TAI = -33 (1er janvier 2006)

2005 Décembre 31	23h 59m 59s
2005 Décembre 31	23h 59m 60s
2006 Janvier 1	0h 0m 00s

jean-philippe.babau@insa-lyon.fr

Gestion de l'heure

- UTC
  - UTC (OP) : erreur maximale de 100 nano secondes
  - Récupérable sur la fréquence porteuse de France Inter
- GPS
  - Liée à l'heure UTC(USNO)
  - Pas d'ajustement avec le TAI: actuellement 19 secondes d'avance
  - Erreur satellite
    - dispersion de 30 ns
    - erreur / UTC : 100 ns heure
    - TAI
- Exemple de synchronisation via le protocole NTP (Net Time Protocol)
  - Évaluation de la charge réseau
  - Évaluation du temps de transmission de l'heure



jean-philippe.babau@insa-lyon.fr

Propriétés des systèmes embarqués temps réel

- Fortes interaction avec le procédé
  - Système continu
  - Évolution du procédé indépendante du contrôle
  - Contraintes temporelles
- IHM limité ou spécifique
  - Parfois pas d'IHM (systèmes enfouis)
  - Modèles d'interactions avec l'utilisateur spécifique
    - Impact sur le développement et les tests
    - Impact sur le fonctionnement
  - Politique d'initialisation, demaintenance
    - Chargement du code, activation, reboot, installation/désintallation
- Contraintes de coût, d'espace, de consommation
  - Matériel spécifique
  - Taille mémoire limitée
  - Processeurs limités

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Propriétés des systèmes embarqués temps réel

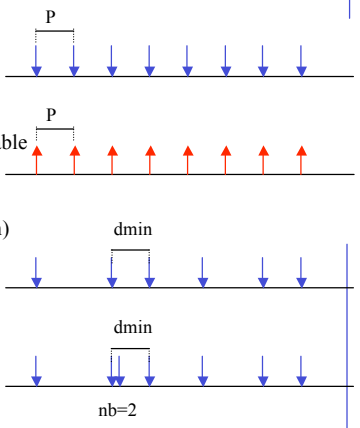
- Prédictibilité
  - Temps : temps d'exécution et temps de réponse
  - Espace : occupation mémoire
  - Energie : utilisation des accès mémoire, vitesse processeur, utilisation des composants
- Fiabilité / sûreté
  - Image de marque
    - Matériel défaillant : matériel non utilisable
    - Image commerciale du vendeur du système
  - système sans IHM
    - séquence de reprise, boot
  - Sécurité
    - Personnes
    - Procédés contrôlés
    - Environnement

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Propriétés temporelles

- Loi d'arrivée d'événements
  - en entrée ou en sortie
  - Périodique (P)
    - Entrée : scrutation périodique, alarme programmable
    - Sortie : stimulation régulière
  - Sporadique
    - intervalle minimum entre deux événements (dmin)
    - rafales (nb, dmin)
  - Apériodique
    - pas d'information temporelle
    - Événement rare
    - Situation d'urgence ou événement anodin



jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

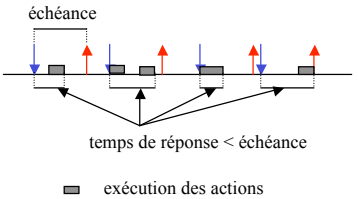
Propriétés temporelles

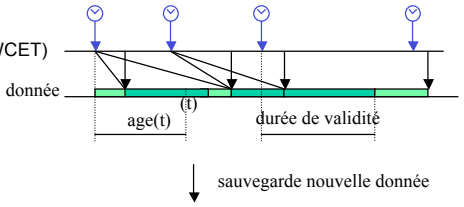
- Échéance
  - relation entre une sortie et une entrée
    - Contrainte sur le temps de réponse maximal
    - Worst Response Time ou WRT
  - échéance bout-en-bout

- Age d'une donnée
  - datation
  - durée de validité
    - Contrainte sur l'age maximal acceptable

- Temps d'exécution
  - Pire cas (Worst Case Execution Time WCET)
  - Temps moyen
  - ≠ du temps de réponse

- Gigue
  - Variation dans un intervalle
  - Période, temps d'exécution, …





jean-philippe.babau@insa-lyon.fr

Systèmes dédiés

- L'architecture matérielle et logicielle est adaptée aux besoins spécifiques de l'application
- Le matériel
  - Peu coûteux, fiable, prédictible
  - Dimensionné pour l'application et son utilisation
    - Capacités de calcul et de stockage
    - Consommation
- Système d'exploitation
  - Limités aux fonctions utilisées par l'application
  - Exemple de Windows XPE
    - boot sur l'application
    - chargement des services nécessaires

jean-philippe.babau@insa-lyon.fr

Bogues célèbres

- Mission vénus
  - passage à 500 000 km (prévu : 5000)
  - remplacement d'une virgule par un point
- Avion F16
  - déclaré « sur le dos » à l'équateur
  - erreur de signe
- Métro de San-Francisco
  - train fantôme
- Lancement 1ère navette américaine
  - faux départ
  - erreur de synchronisation entre les deux ordinateurs

jean-philippe.babau@insa-lyon.fr

Bogues célèbres

- Robot d 'exploration de Mars
  - reset général
  - mauvaise politique d 'ordonnancement
- Problème de modélisation de l'environnement
  - Avion
    - " pas de rétro-propulsion sur un avion en l'air "
    - "Avion au sol" = " roue sorties et roues qui roulent "
    - Piste verglacée …
  - Voiture
    - " pas de verrouillage lorsque le véhicule est occupé "
    - " véhiculé occupée " = " clé électronique proche "
    - Au garage, la clé doit être loin …

jean-philippe.babau@insa-lyon.fr

Bogues célèbres : Ariane 5

- Contexte
  - Le calculateur SRI mesure l'attitude du lanceur et ses mouvements (angles et vitesse).
  - Les données du SRI sont transmises, via le bus de données, au calculateur embarque (OBC) qui exécute le programme de vol et qui commande les tuyères et les moteurs
  - Redondance : deux SRI travaillent en parallèle ; ces systèmes sont identiques tant sur le plan du matériel que sur celui du logiciel. L'un est actif et l'autre est en mode "veille active" ; si l'OBC détecte que le SRI actif est en panne, il passe immédiatement sur l'autre SRI a condition que ce dernier fonctionne correctement
  - La conception des SRI d'Ariane 5 est pratiquement la même que celle d'un SRI qui est actuellement utilise a bord d'Ariane 4, notamment pour ce qui est du logiciel.
- Bogue
  - braquage des tuyères l'OBC agissant sur la base des données transmises par le SRI
  - Une partie des données ne contenait pas des données de vol proprement dites mais affichait un profil de bit spécifique de la panne du calculateur du SRI 2 qui a été interprété comme étant des données de vol
  - Le SRI actif avait déclaré une panne due a une exception logicielle et le SRI de secours avait déjà cessé de fonctionner durant le précédent cycle de données (période de 72 millisecondes) pour la même raison

jean-philippe.babau@insa-lyon.fr

Bogues célèbres : Ariane 5

- Bogue
  - l'exception logicielle est due à une conversion d'un flottant sur 64 bits en entier sur 16 bits (dépassement de capacité et code Ada)
  - pas de protection sur cette exception
  - l'erreur s'est produite dans une partie du logiciel utilisée uniquement avant le décollage
  - l'erreur d'opérande est due à une valeur élevée d'une variable appelée BH (Biais Horizontal) et liée à la vitesse horizontale de la fusée. La valeur BH était nettement plus élevée que la valeur escomptée car la première partie de la trajectoire d'Ariane 5 diffère de celle d'Ariane 4, ce qui se traduit par des valeurs de vitesse horizontale considérablement supérieures
- Analyse
  - les événements internes du SRI qui ont conduit a l'accident ont été reproduits par simulation
  - les deux SRI ont été récupérés pendant l'enquête et le contexte de l'accident a été détermine avec précision a partir de la lecture des mémoires

jean-philippe.babau@insa-lyon.fr

Bogues célèbres : Ariane 5

- Erreur
  - Erreur de codage
    - Pas de protection sur exception non prévues
  - Erreur de spécification
    - Domaine d'entrée des variables d'Ariane 4 au lieu d'Ariane5
    - Modélisation du procédé
  - Erreur de tests / mise au point
    - Pas d'émulation des conditions de vol d'Ariane5
    - Pas de test d'intégration
      - désormais obligatoire
  - Erreur de conception
    - Une exception n'est pas une erreur grave
    - Un code inutile doit être arrêté ou confiné

→ Maîtrise du développement

jean-philippe.babau@insa-lyon.fr

Besoins de fiabilité

- Automobile
  - Image de marque (régulateur de vitesse)
  - Coût d'un retour de séries
  - Aspect psychologique du système enfoui
- Objets personnels
  - Rejet car « ne marche jamais »
  - Reset = perte de données !
  - Doit être accessible à tous, sans opération de configuration
  - Partagé par toute la famille
- QoS
  - image numérisée = image « parfaite »
  - Retard trop important = panne du point de vue de l'utilisateur
  - Consommation énergétique maîtrisée
- Durée de vie du système
  - Voiture : 10 ans, train : 20 ans
- Contractualisation
  - Vendeur du produit final : « la marque »
  - Intégrateur, sous-traitant

jean-philippe.babau@insa-lyon.fr

Développement

- Modèle économique : plusieurs acteurs
  - Intégrateur / vendeur
    - Automobile : constructeur
    - Multimédia : opérateur
  - Fournisseur
    - Automobile : équipementier
    - Multimédia : développeur de produits, prestataire de services
  - Fournisseur de technologies
    - Fondeurs, RTOS
  - Contractualisation des rapports
  - Peu d'intégration
- Cycle de développement
  - cycle en V : identification d'étapes, séparation client / fournisseur
  - approche « maison »
    - réutilisation généralement limitée aux fonctions

jean-philippe.babau@insa-lyon.fr

Développement

- Etapes
  - Spécification
    - Expression des besoins fonctionnels et contraintes matérielles
    - Description du procédé à contrôler, de l'environnement
    - Contraintes de qualité de service (coût, taille, consommation)
  - Conception
    - Découpage matériel / logiciel
    - Architecture matérielle
      - Cartes, composants
      - Traitements et protocoles à spécifier
    - Architecture logicielle
      - OS, environnement de développement
      - Décomposition modulaire
  - Codage
    - Choix d'un environnement de développement et d'un langage (C)
  - Validation

jean-philippe.babau@insa-lyon.fr

Déroulement d'un projet (vue du développeur informatique)

1. Spécifications des besoins

2. Choix d'une architecture matérielle : processeur (puissance, famille), carte

- Puissance de calcul nécessaire (x MIPS ...)
- Besoins en communication et entrées/sorties

- Choix a priori dans 75% des cas
  - énergie, prix
  - besoins spécifiques
  - Domaine d'application
  - culture d'entreprise
  - coût du changement (formation, prise en main)

jean-philippe.babau@insa-lyon.fr

3

Choisir un langage de programmation
et un OS (RTOS), si nécessaire

- Plus de 70 SE disponibles
- Disponibilité vis-à-vis de la plateforme
  - Processeur supporté ?
  - Drivers pour la carte à développer ?
- Coût
  - Plateforme de développement, intégration de services, royalties
- Domaine
  - Ferroviaire : QNX
- Documentation et fiabilité des fournisseurs sur le long terme

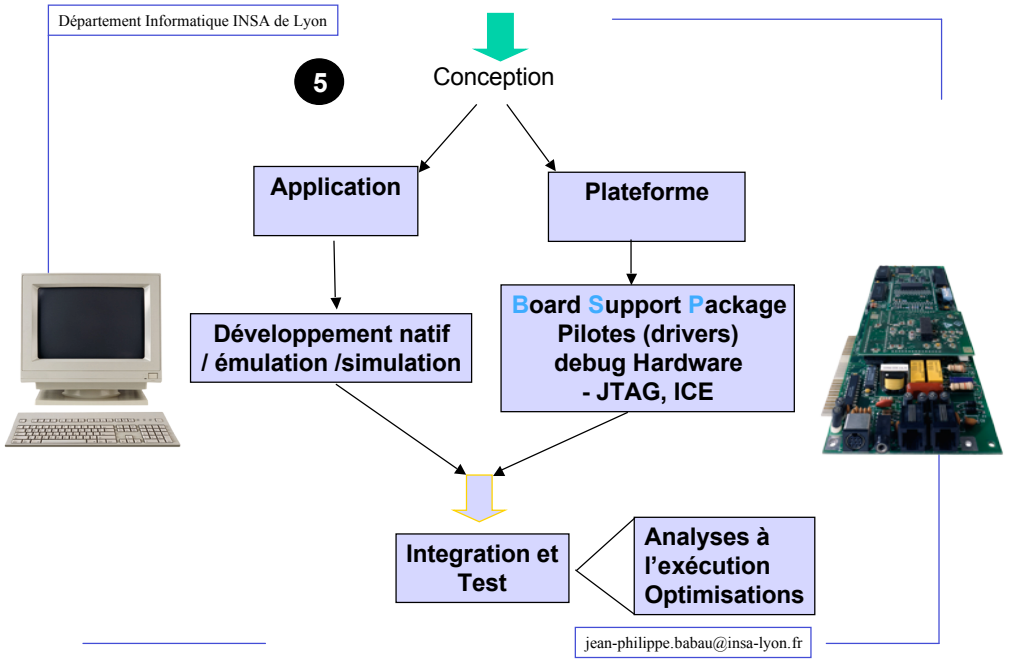
jean-philippe.babau@insa-lyon.fr

4

Choix de l'outil de développement
(compilateur, suite de développement, debugger, simulateur)

- Dépend généralement de l'OS et du langage
- L'utilisation d'Eclipse est en pleine explosion

jean-philippe.babau@insa-lyon.fr



jean-philippe.babau@insa-lyon.fr

Méthodes, technologies et langages

- Langages de spécification et de modélisation
  - **SART**, textuel, Matlab, StateMate, *logiques (RealTime Logic)*
  - UML (paradigme objet peu répandu), profils UML (MARTE)
  - SDL pour les télécoms (-> UML 2.0)
  - Méthodes et langages formels (B)
- Langages de conception
  - **UML**, SDL, codesign, ADL et Composants
  - multitâche (**LACATRE**)
  - approches synchrones (Esterel, Signal, Lustre)
- Ingénierie dirigée par les modèles (IDM)
  - Model Driven Engineering (MDE)
  - Modèles , métamodèles (concepts), transformation de modèles

jean-philippe.babau@insa-lyon.fr

Méthodes, technologies et langages

- Langages d'action
  - Assembleur, C
  - Java, « XX Java » (**JavaCard,** RealTimeJava, EmbeddedJava)
- Systèmes d'exploitation
  - exécutif maison
  - système temps réel RTOS (**VxWorks**)
  - système généraliste spécialisé (RTX, CE, RTLinux)
  - Machines virtuelles (EmbeddedJava, Real-TimeJava)
- Environnements de développement
  - Spécifique à la carte et au langage
    - éditeur, compilateur, éditeur de lien et *mapping* mémoire spécifiques
  - Mise au point
    - instrumentation du code (printf sur LCD, led, …)
    - émission sur un port série
    - debug via la liaison JTAG

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

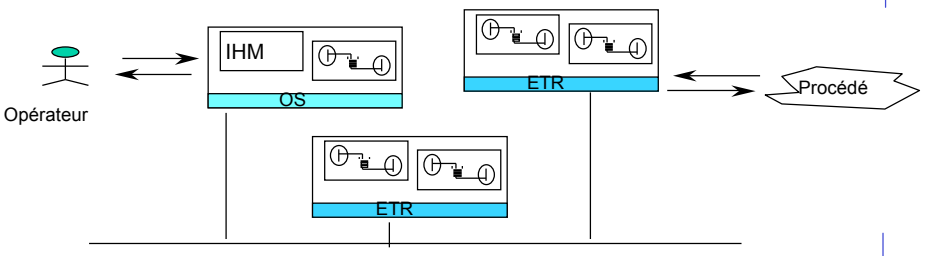
Méthodes, technologies et langages

- IHM limité, spécifique ou inexistant
- Communication
  - bus(VME), liaison série
  - réseaux de terrain (I2C, **CAN**, TTP, FlexRay, FIP, ARINC, AFDX)
  - réseaux sans fil (irDA, Bluetooth)
  - protocoles classiques (socket, TCP/IP)
- Vérification / validation
  - preuves, test exhaustif, simulation
  - **ordonnancement**, logique temporelle
  - simulation réaliste
  - certification de code (avionique) ou de compilateur
    - règles d'écriture

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Systèmes complexes



- Système hétérogène
  - temps réel / non temps réel
  - critiques / non critiques
- Système communicants
- Système évolutif / adaptatif
  - ajout / suppression / modification
  - modes d'urgence
  - garantie de QoS

jean-philippe.babau@insa-lyon.fr

Département Informatique INSA de Lyon

Les défis actuels

- Maîtrise des systèmes complexes
- Portabilité des applications
  - Standardisation des plateformes d'exécution et de communication
- Intégration des nouveauté technologiques
  - Legacy code
- Réutilisation
  - Savoir-faire : composants métier
  - Composants à l'exécution
- Partage de services, sûr de fonctionnement
  - Contractualisation

jean-philippe.babau@insa-lyon.fr

Objectifs du cours

- Analyser un problème
  - Langage SA-RT
- Connaître les plateformes
  - Notions d'architecture matérielle
  - Choisir et implémenter un RTOS
  - Connaître les protocoles de communication pour l'embarqué
  - Découverte d'une plateforme spécifique : la javaCard
- Concevoir
  - Programmation multitâches
  - Concevoir et programmer un pilote intégré et sa couche service
- Faire une analyse temps réel (RMA)

jean-philippe.babau@insa-lyon.fr