

Spécification et conception conjointe des systèmes matériel/logiciel

Méthodologie et outil

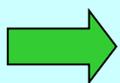
Jean-Paul CALVEZ

Equipe de Recherche
« Modélisation et Conception des Systèmes Electroniques »

Polytech ' Nantes, France

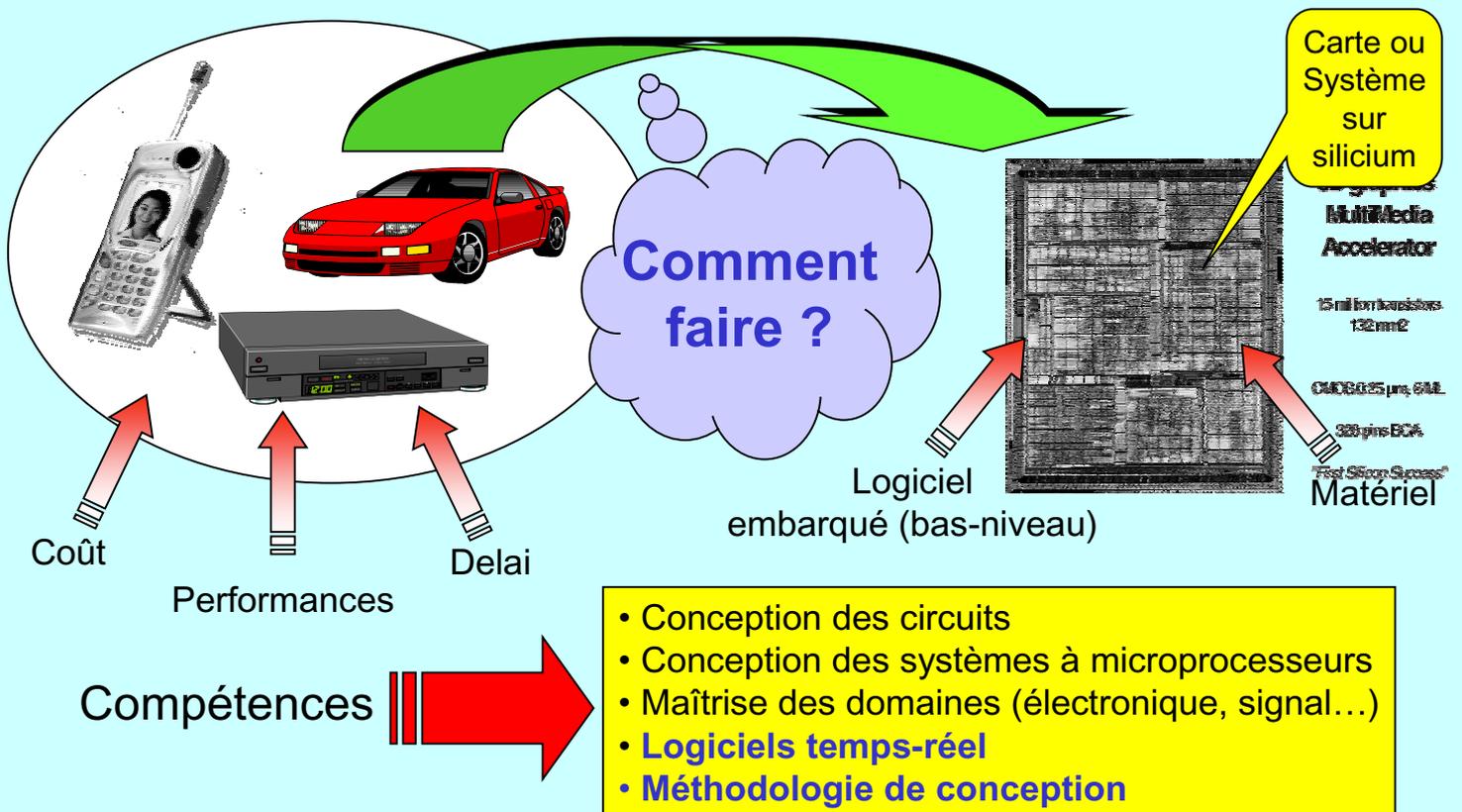
jean-paul.calvez@polytech.univ-nantes.fr

Plan



- **La conception système: c'est quoi**
- Les bases de la modélisation et de la conception
- Spécification d'un système
- Conception fonctionnelle
- Conception architecturale
- Un outil comme support
- Conclusions

Concevoir : c'est quoi?



Quelques définitions (1/2)

- **Système:**
 - Toute "chose" constituée de parties organisées pour assurer une fonction ou un ensemble de fonctions dans son environnement
- **Architecture:**
 - La structure fondamentale et unifiante du système définie en terme de ses constituants, interfaces, liens, comportements et contraintes
 - basée sur un ensemble de concepts
 - décrite par un ensemble de vues (perspectives) liées entre elles
- **Propriétés émergentes:** propriétés résultant du couplage entre les constituants et pas uniquement de chaque composant (+ que somme des parties)
 - propriétés observés, voulues, exigées
 - propriétés inattendues, interdites...
 - Importance de l'environnement (incertain, réactif.....)

Quelques définitions (2/2)

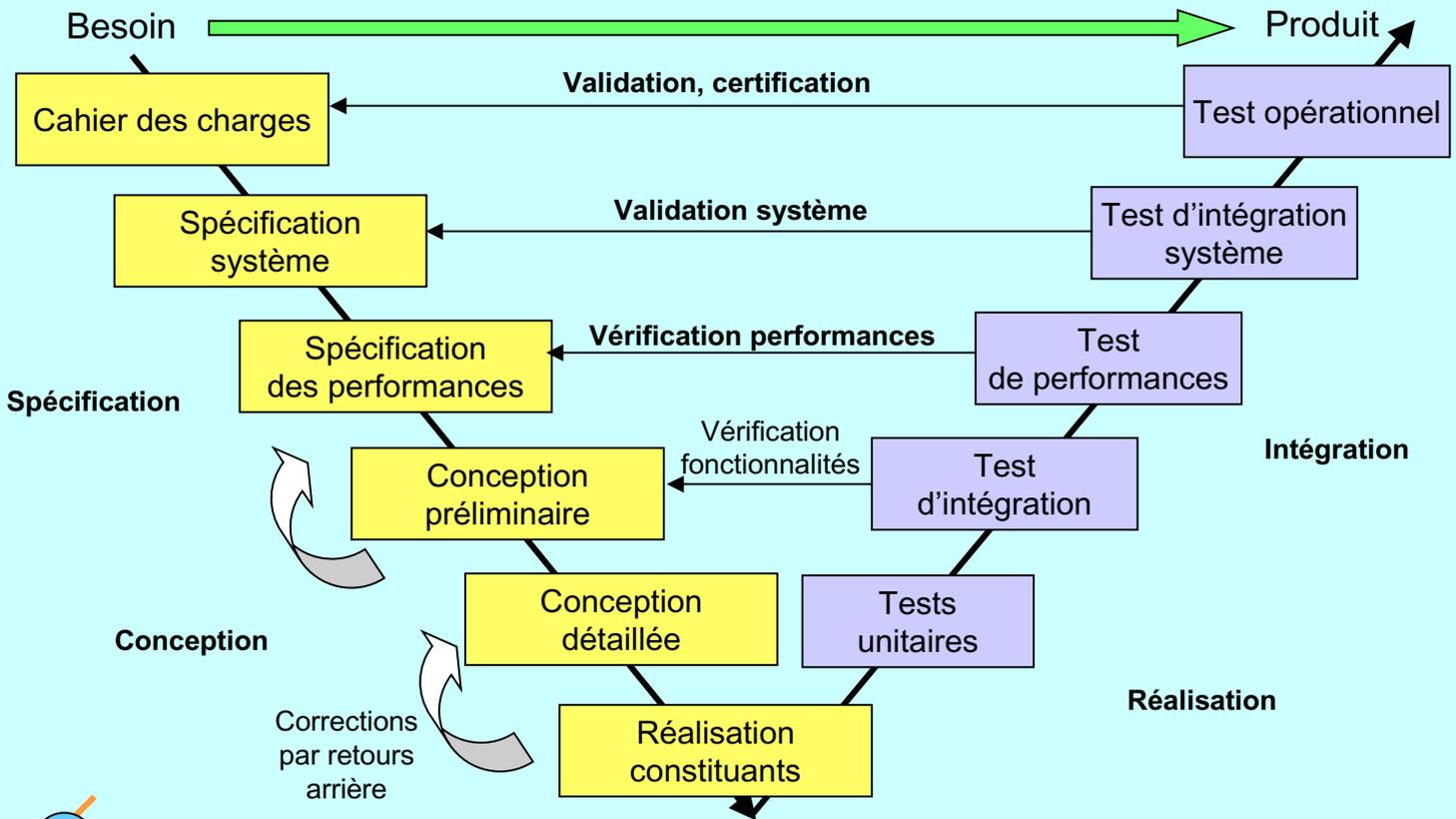
- **Méthode:**
 - technique de résolution de problème caractérisée par un ensemble de règles bien définies qui conduisent à une solution correcte
- **Méthodologie:**
 - un ensemble structuré et cohérent de modèles, méthodes, guides et outils permettant de déduire la manière de résoudre un problème
- **Modèle:**
 - une représentation d'un aspect partiel et cohérent du « monde » réel
 - précède toute décision ou formulation d'une opinion
 - est élaboré pour répondre à la question qui conduit au développement du modèle
 - n'est donc pas une fin en soi

Rôle d'un modèle pour les systèmes

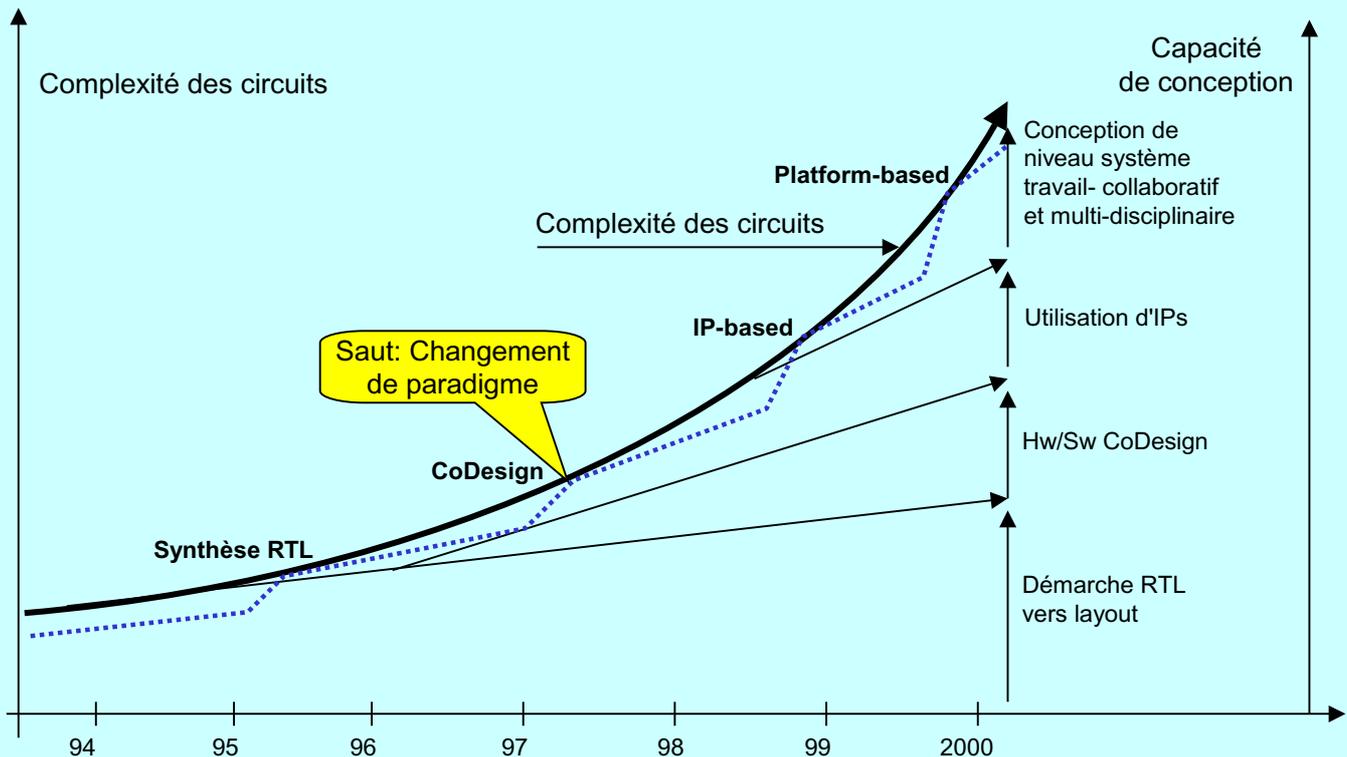
- **Abstraction**
 - Eliminer des détails, focaliser sur un point de vue du système
 - Travailler à différentes échelles de complexité et de temps
- **Analyse**
 - Etude des propriétés du modèle (vérification de propriétés)
 - Extrapolation au système réel représenté
- **Communication**
 - Discussion et échanges avec d'autres personnes
 - Echanges entre outils
- **Génération/Production**
 - Produire une représentation d'un autre niveau (autre modèle)
 - Produire le système réel

=> Modèle à retenir: fonction de l'objectif visé

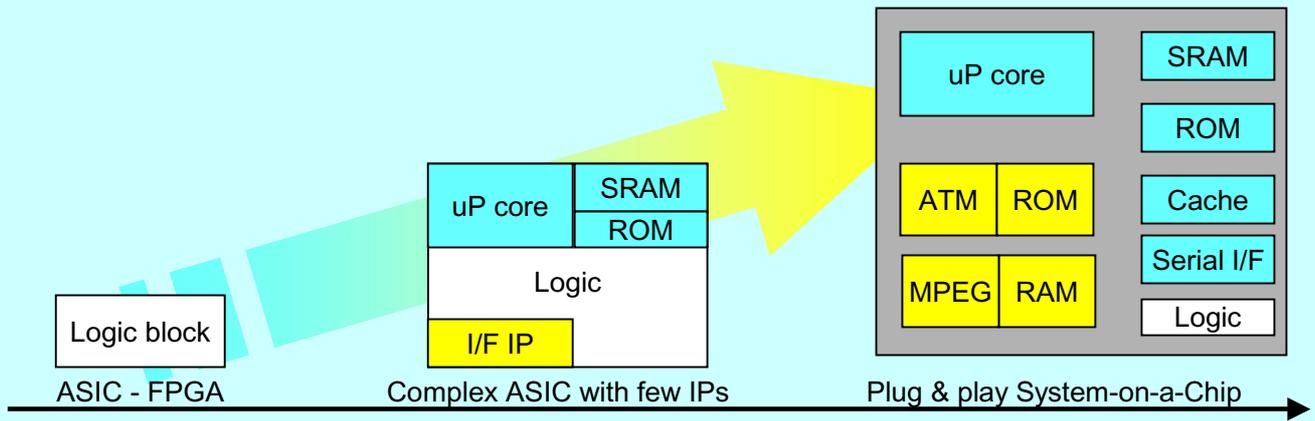
Modèle de cycle de développement en V



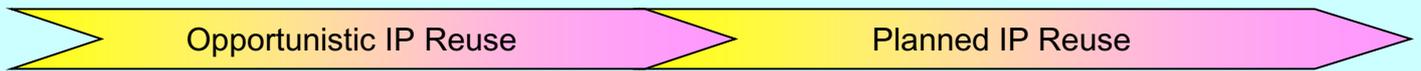
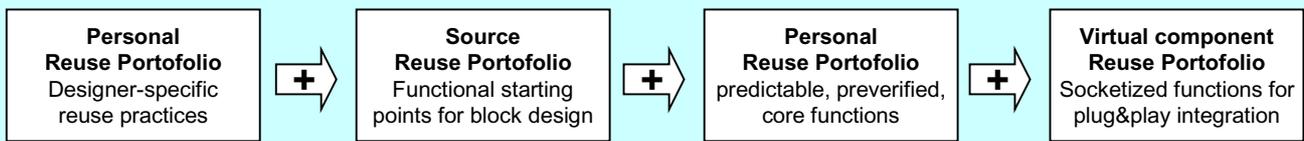
Constat aujourd'hui: Technology/Design Gap



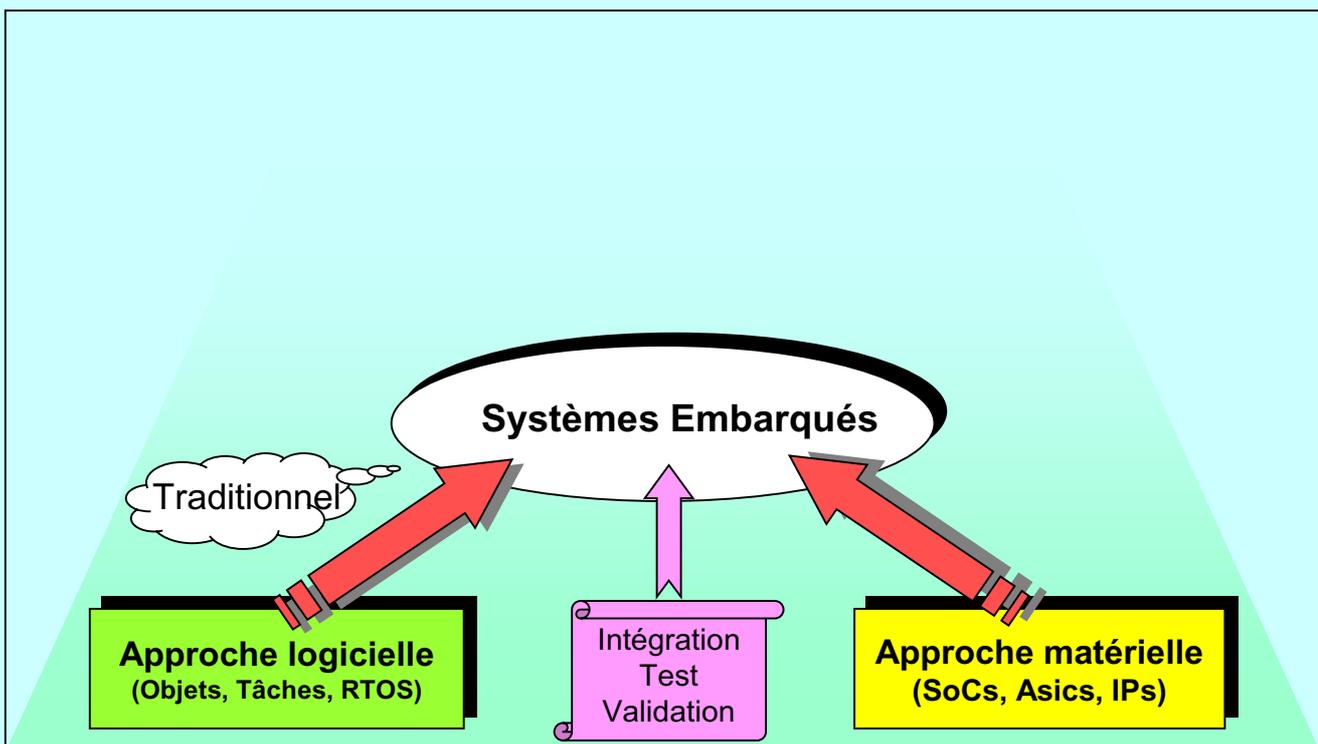
Evolution des méthodologies coté circuits



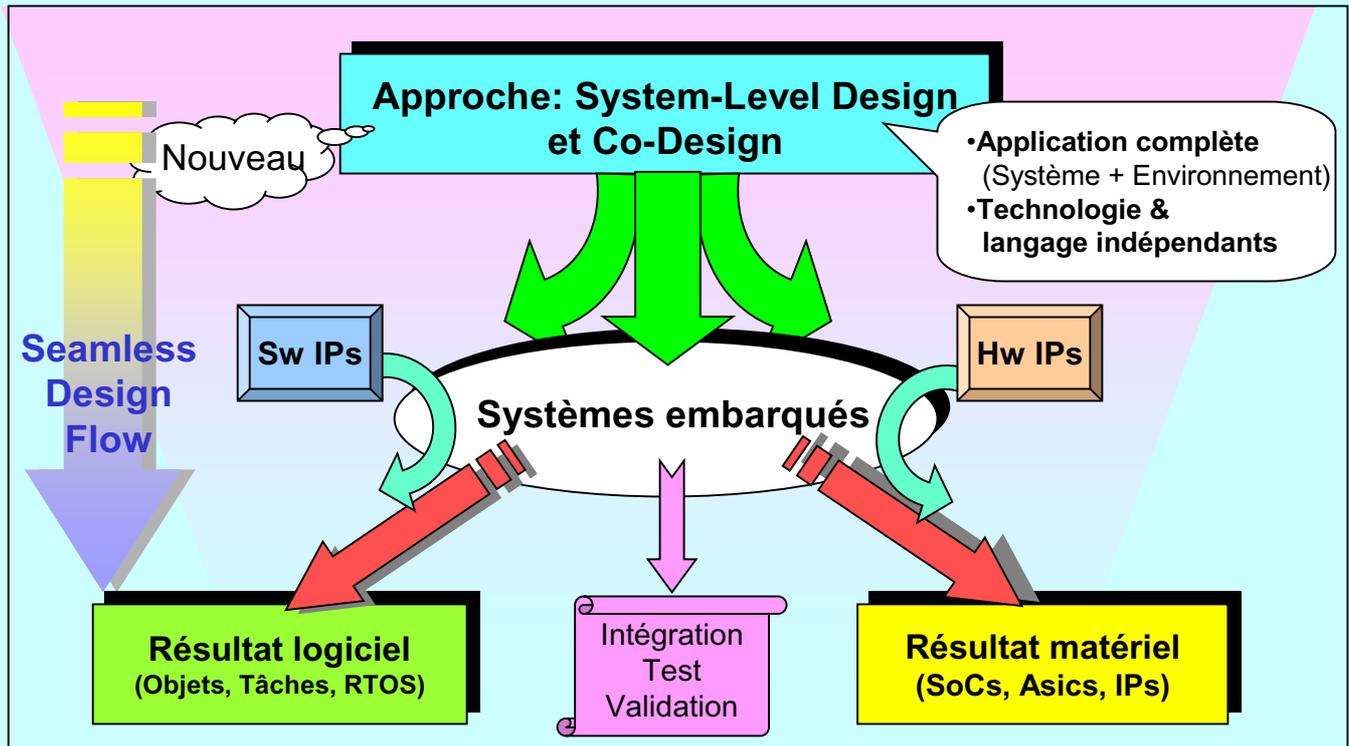
Timing-Driven Methodology (TDD) → Block-Based Methodology (BBD) → Platform-Based Methodology (PBD)



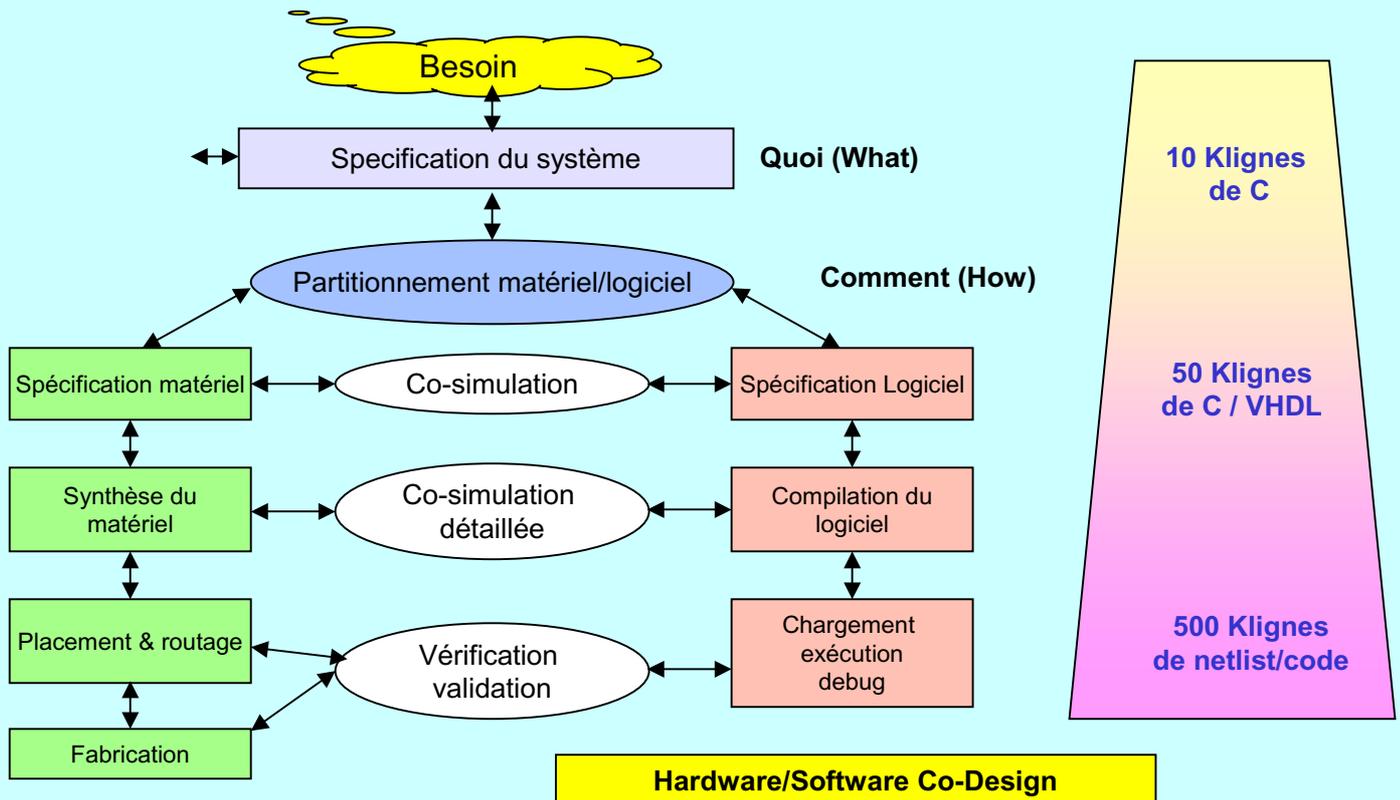
Développer les systèmes différemment (1/2)



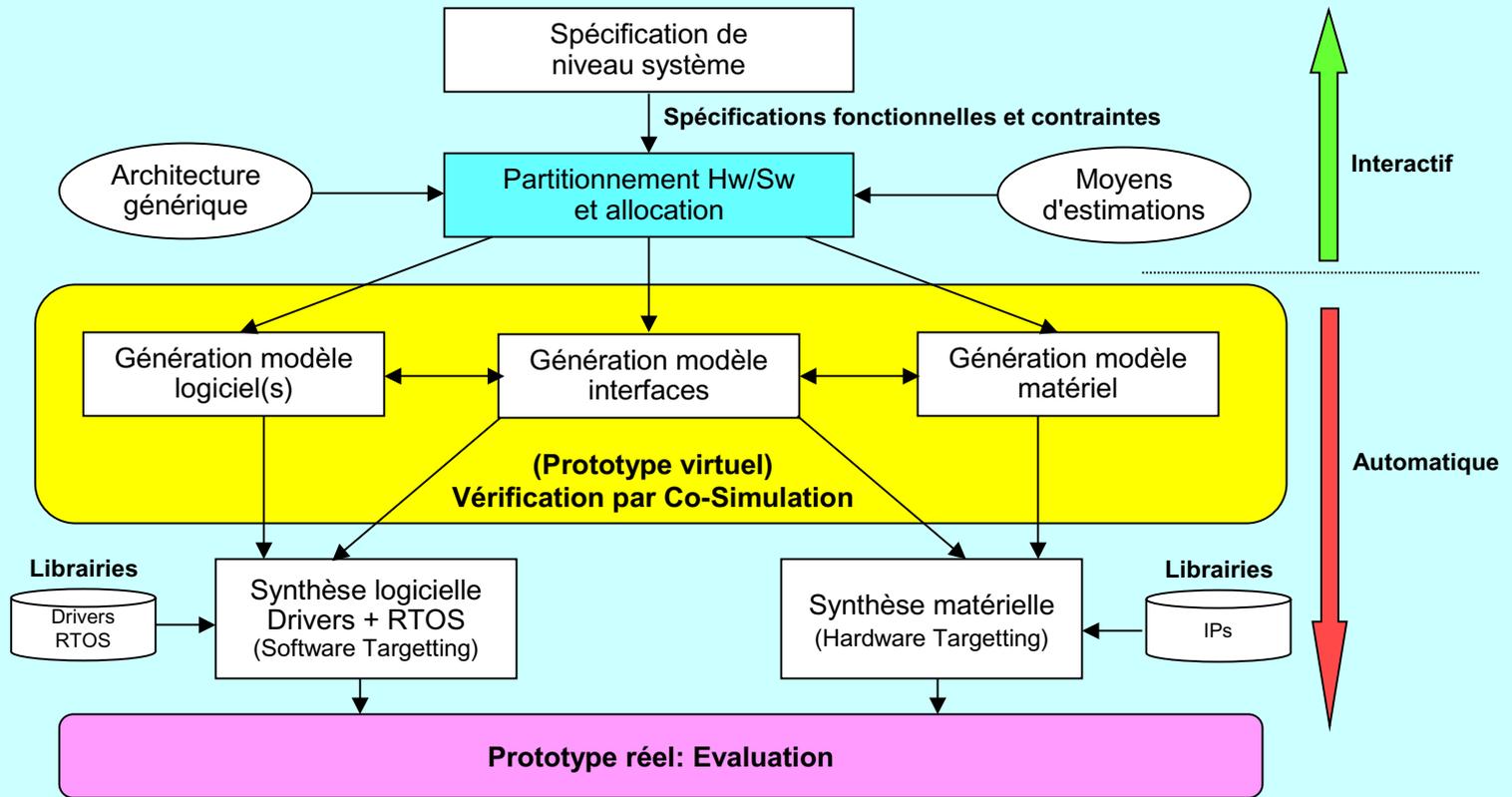
Développer les systèmes différemment (2/2)



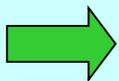
Flot de Conception Aujourd'hui



Démarche de CoDesign

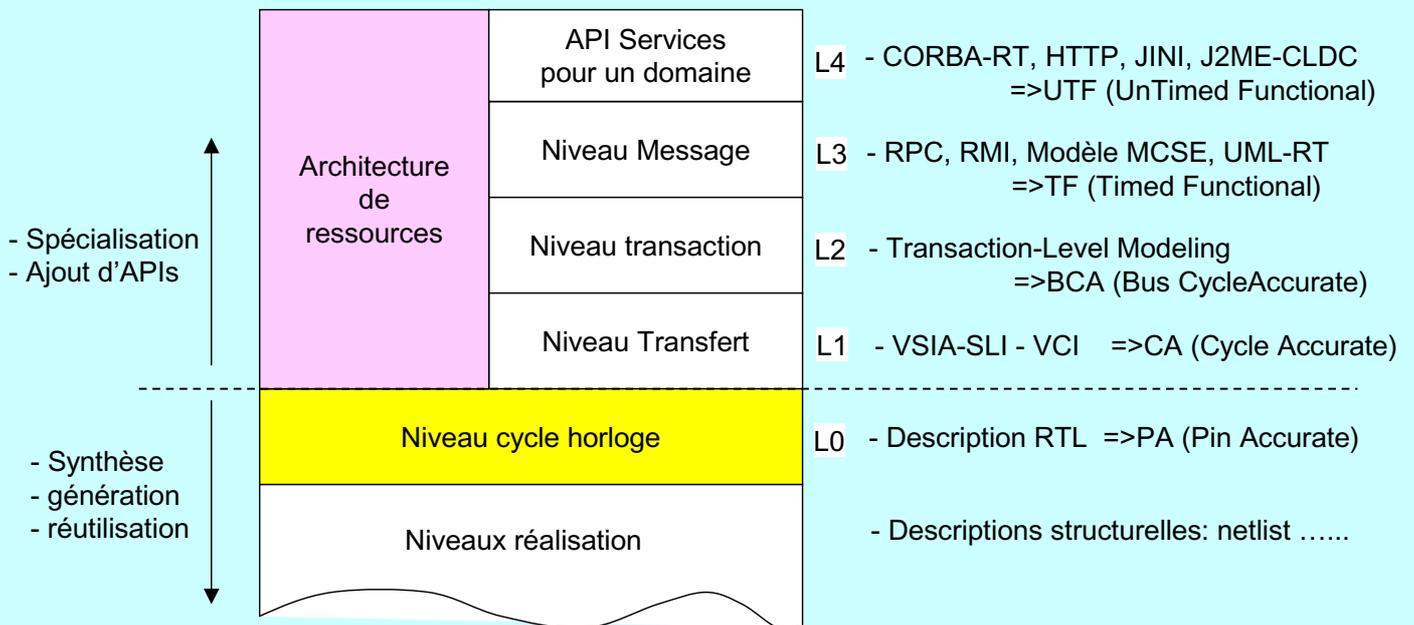


Plan

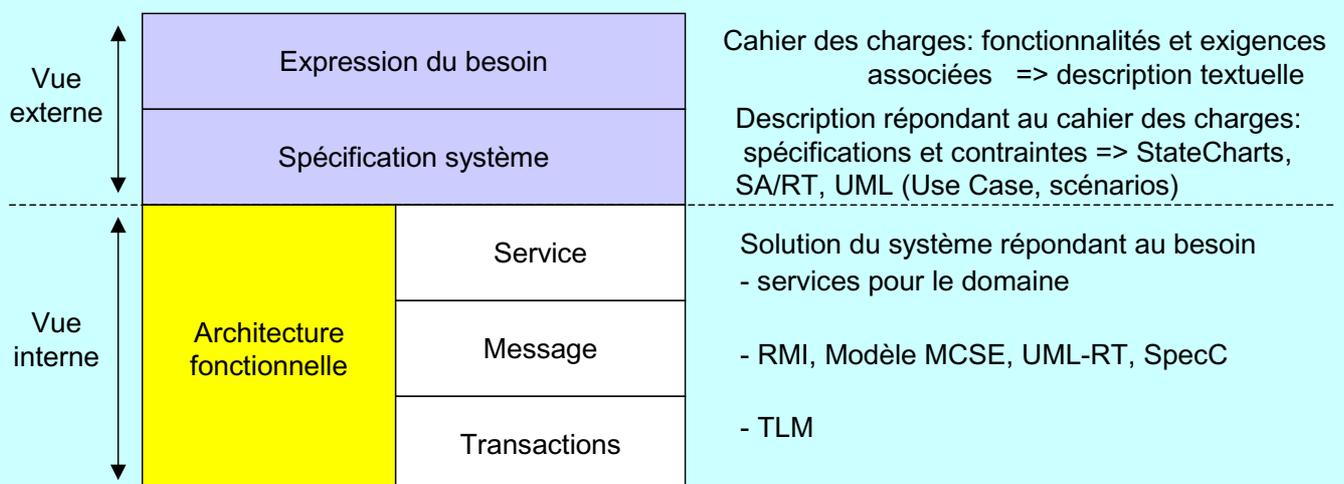


- La conception système: c'est quoi
- **Les bases de la modélisation et de la conception**
- Spécification d'un système
- Conception fonctionnelle
- Conception architecturale
- Un outil comme support
- Conclusions

Niveaux d'abstraction d'une plate-forme



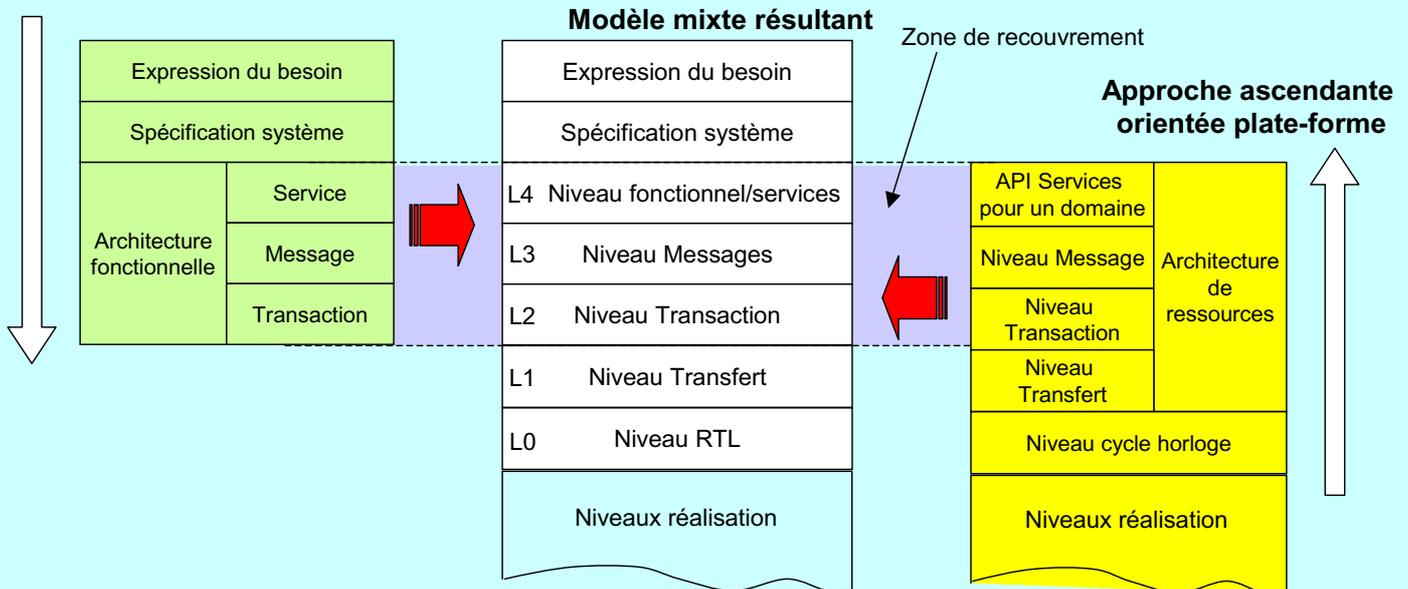
Niveaux d'abstraction d'une application



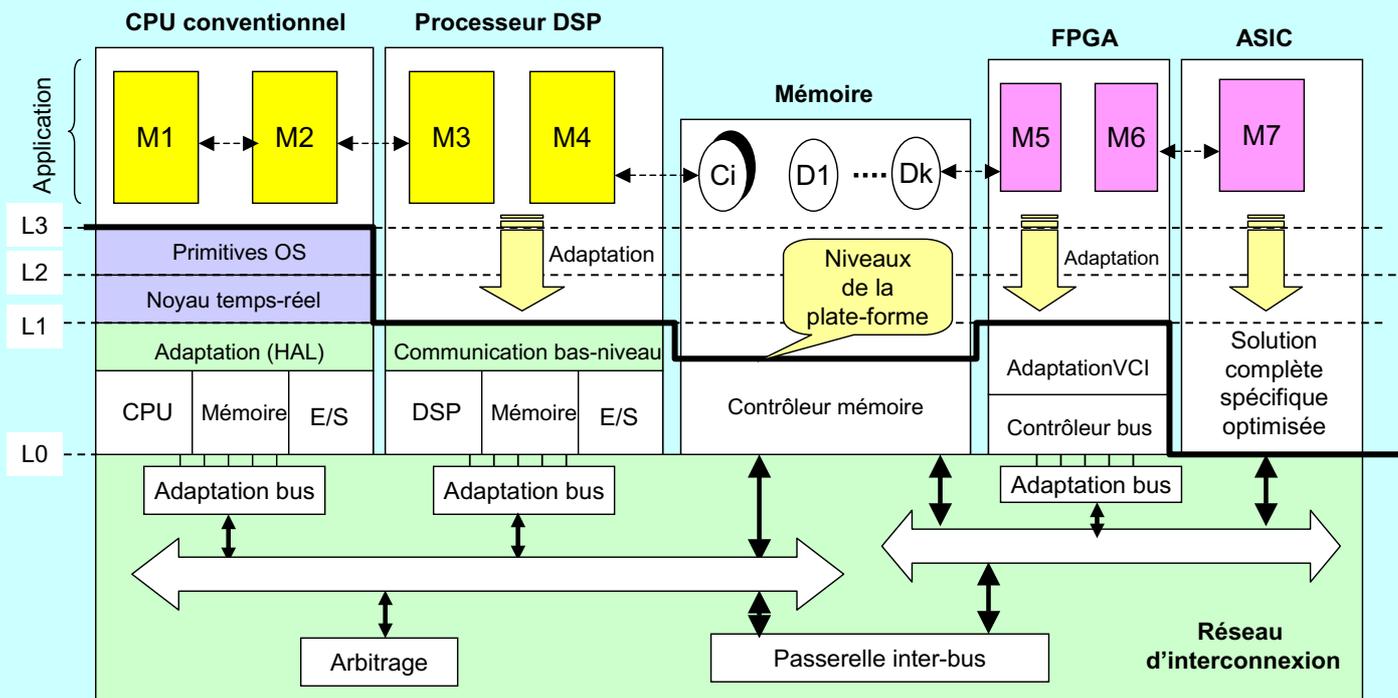
Critère d'arrêt: Dépendance par rapport à la technologie de réalisation

Intégration des 2 points de vue

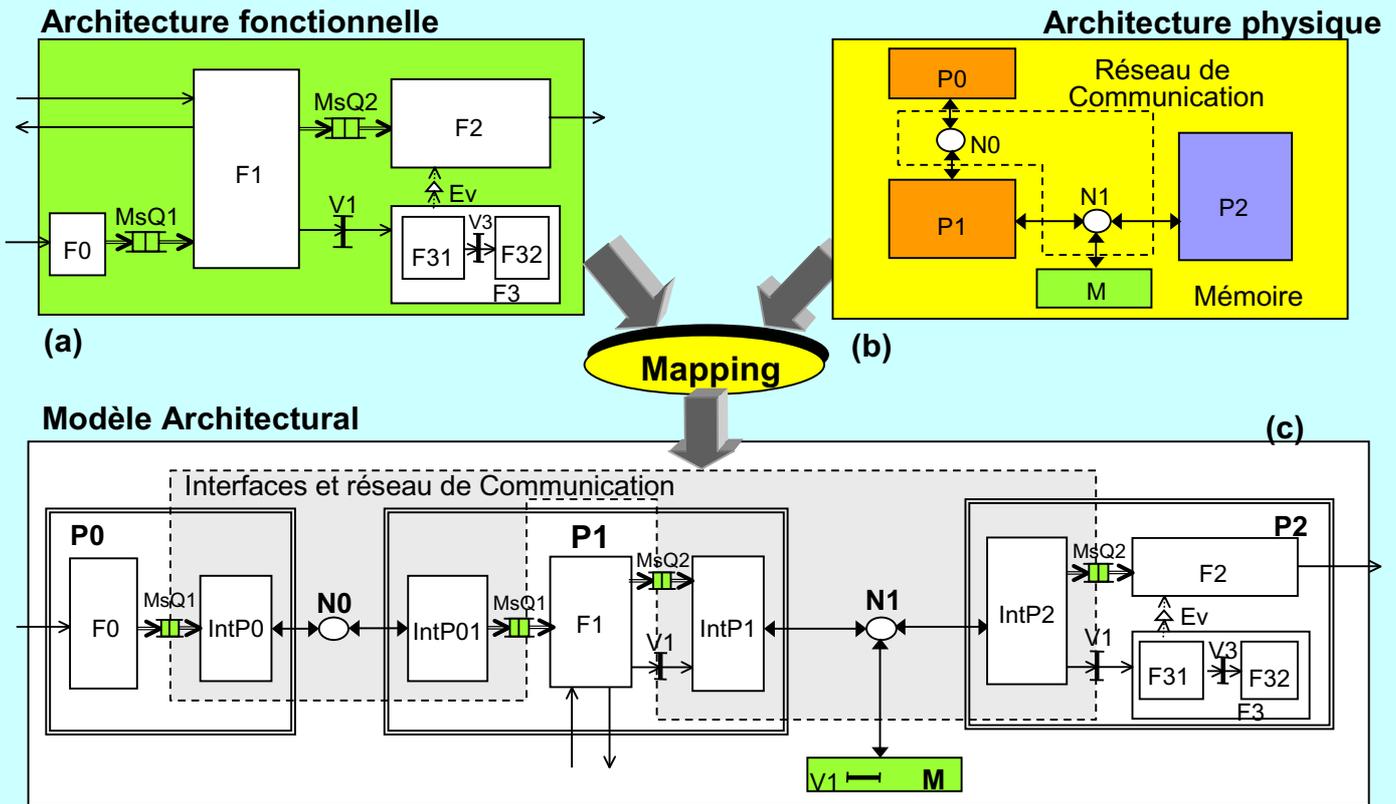
Approche descendante dirigée par l'application



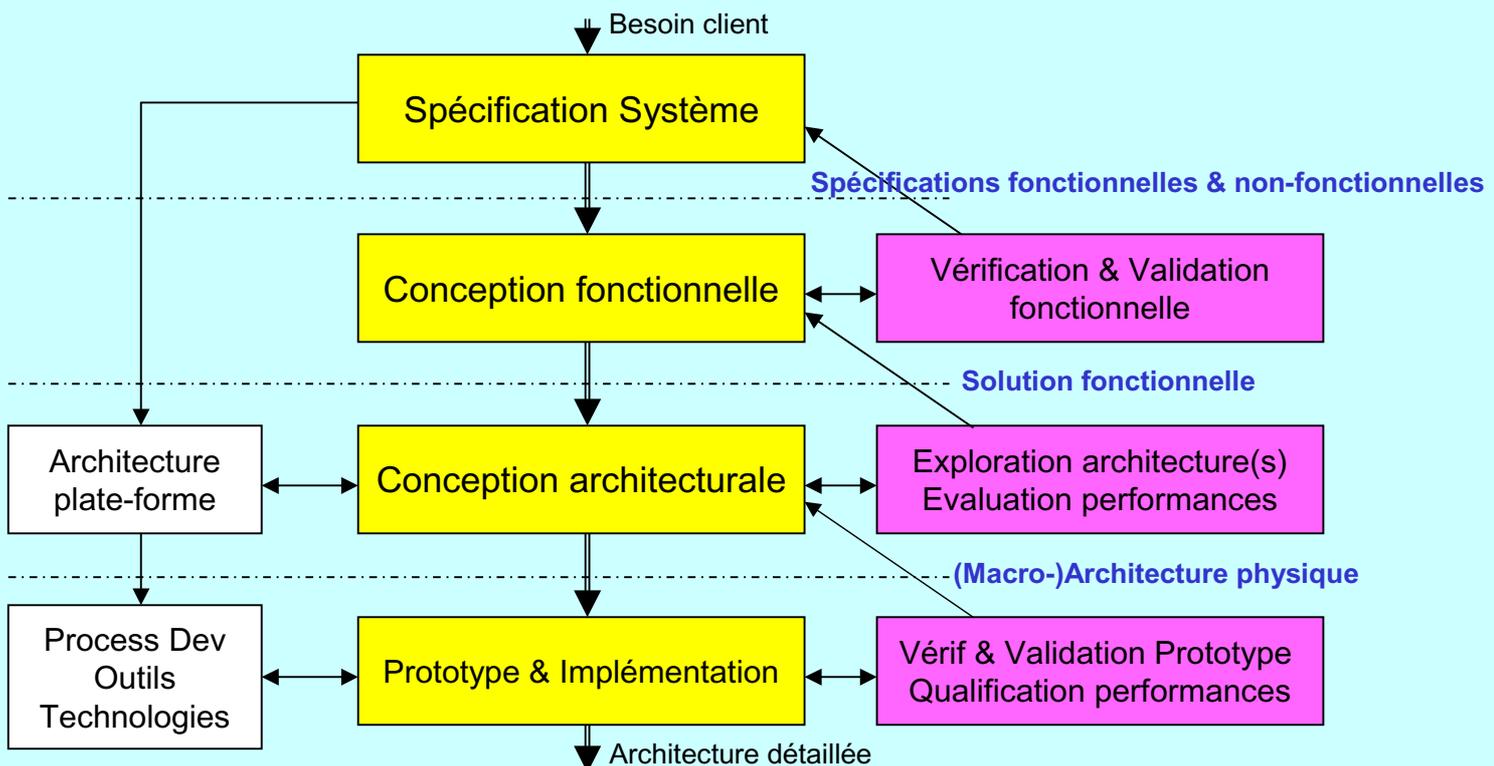
Exemple d'une architecture hétérogène



Modèle en Y



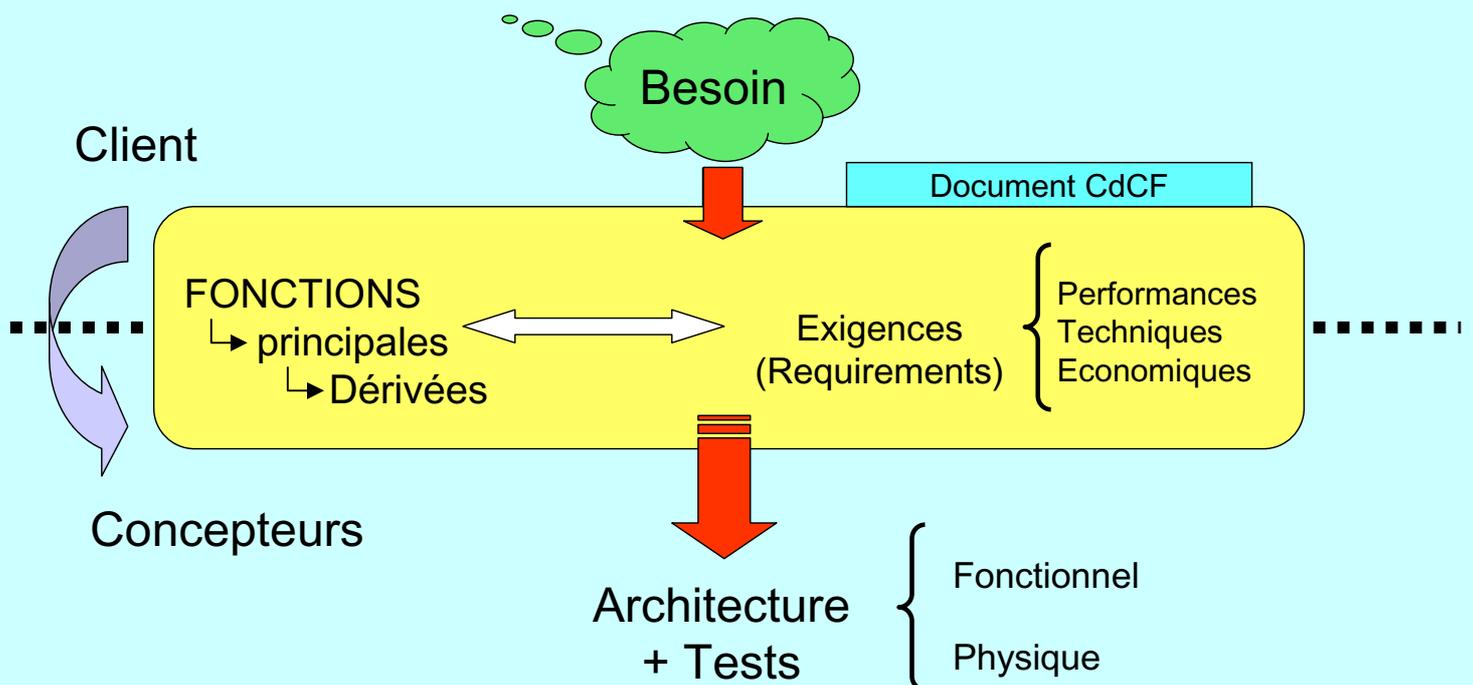
Flot de conception



Plan

- La conception système: c'est quoi
- Les bases de la modélisation et de la conception
- ➔ • **Spécification d'un système**
- Conception fonctionnelle
- Conception architecturale
- Un outil comme support
- Conclusions

Expression des Exigences Système



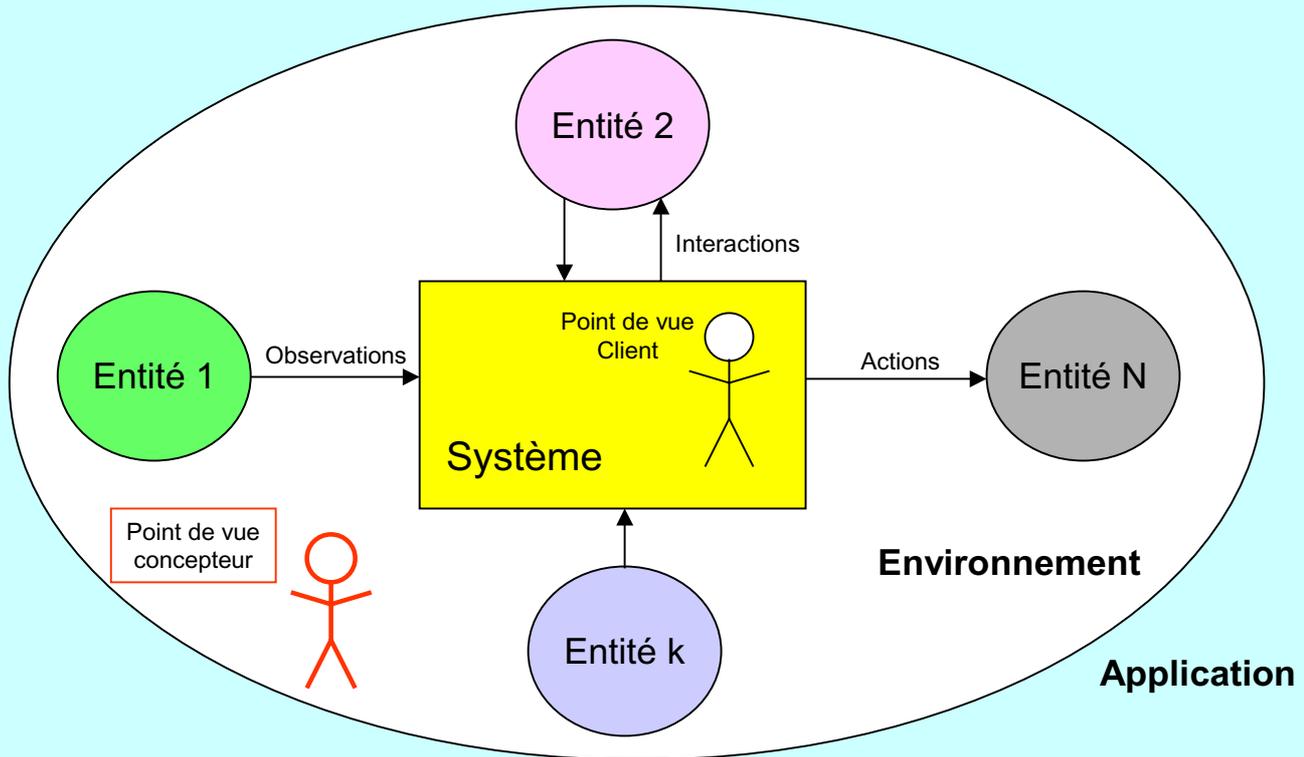
Catégories d'exigences non-fonctionnelles

- Performances
 - performances temporelles: durée, latence, débit,
 - Ressources nécessaires: énergie, mémoire, puissance de calcul ...
 - Capacité de traitement, de communication
- Sûreté de fonctionnement
 - Fiabilité, Sécurité, innocuité, maintenabilité, disponibilité
- Ergonomie
 - Qualité des échanges
 - Compétences d'exploitation
- Contraintes
 - Physiques: dimensionnement, masse, environnement,
 - Electriques, conformité à des normes,
 - Evolutivité, coût et délai, livraison, qualification,

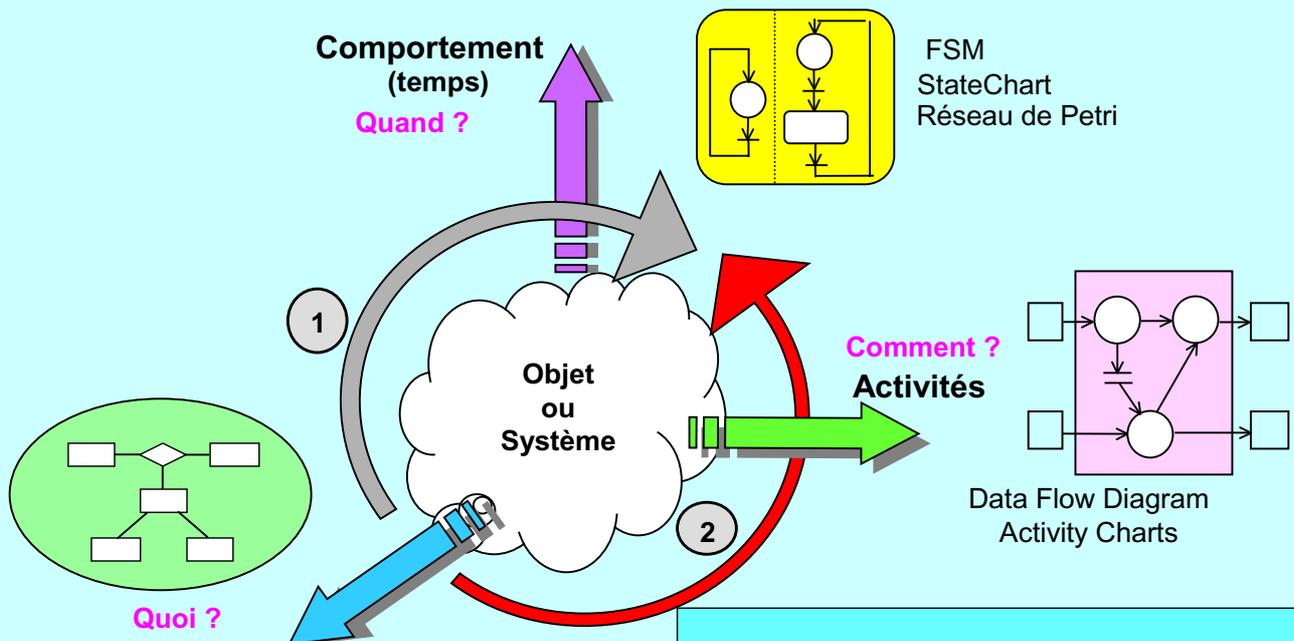
Etape 1 - (Co-)Spécification

- Modèles de spécification
 - Une grande variété de modèles de spécification
 - (Statechart, SDL, SpecChart, MatLab, MatriXx, C, VHDL, SystemC.....);
 - Modèle unique (Lequel?) ou modèle hétérogène multi-langage (Lesquels et couplages?)
 - Difficultés d'exprimer tous les points de vue et les spécifications non-fonctionnelles (contraintes)
- Techniques de vérification
 - Simulation homogène: nécessité d'un simulateur ou conversion en un langage donné
 - StateMate, MCSE, SystemC....
 - Simulation hétérogène:
 - Ptolemy: encapsulation de modèles "d'univers différents" (Classes C++)
 - Couplage de simulateurs par protocole TCP/IP ou sockets (VCI)

Le système à spécifier et son environnement



Approche pour la spécification Système



Objets / Données / Événements Informations

- Modèle hiérarchique de composition de données
- Modèle Entité/association
- Modèlèl Objet (OMT, OOA)

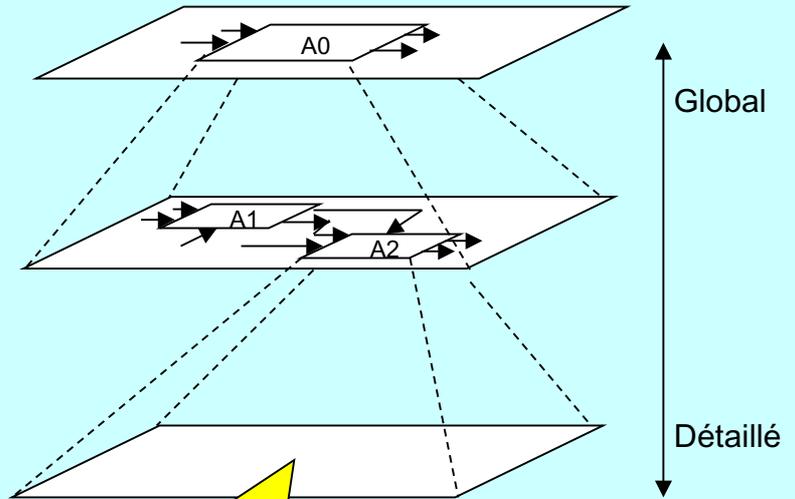
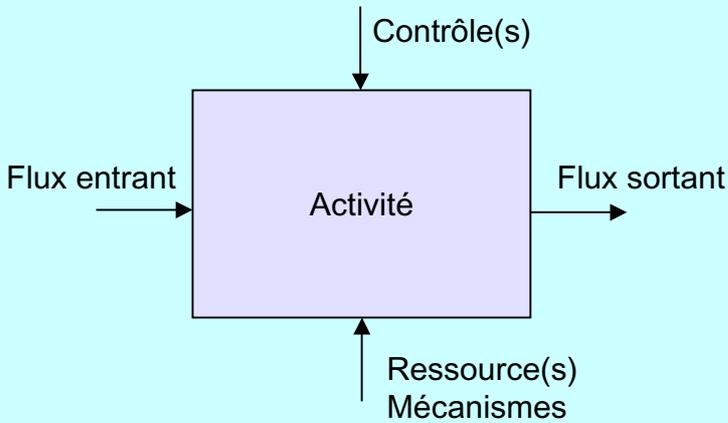
SYSTEM-LEVEL MODELING ?

Notion de TEMPS essentiel pour les systèmes RT

Analyse fonctionnelle

- **Modèle SADT (Ross-77)**

- Actigramme
- Datagramme



Limiter la décomposition à 5 à 7 éléments

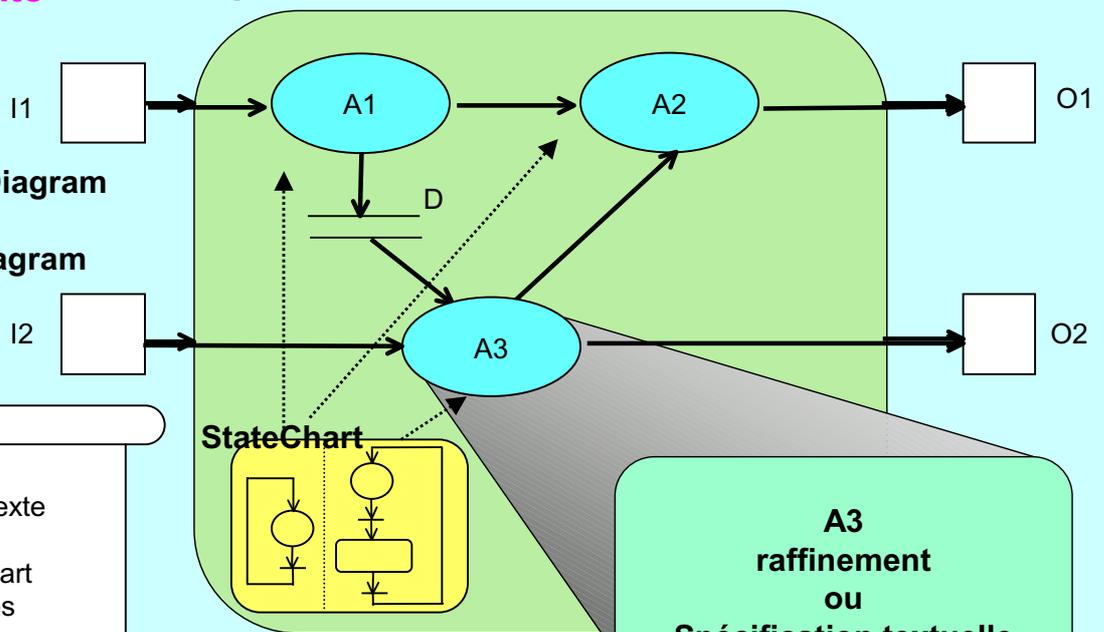
Outil SADT ou IDEF0

Spécification SA/RT (Ward/Mellor)

Outil StateMate (Ilogix)

Data Flow Diagram ou Activity diagram

Diagramme de Contexte



Méthode:

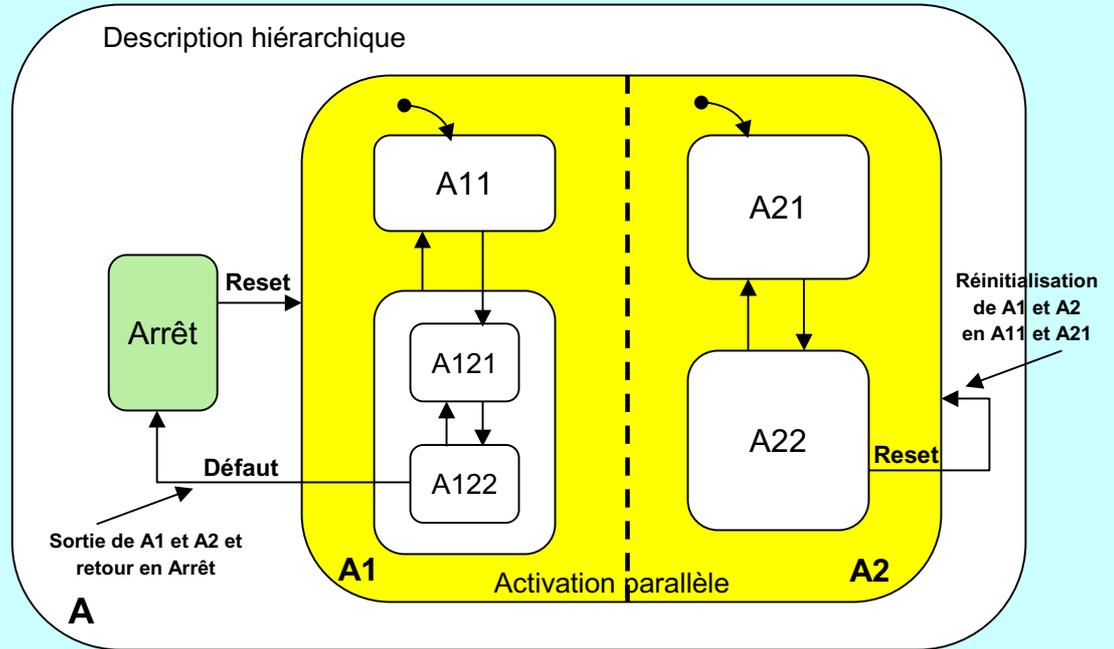
- diagramme de contexte
- décomposition DFD
- introduction Statechart
- spécification activités

Difficultés

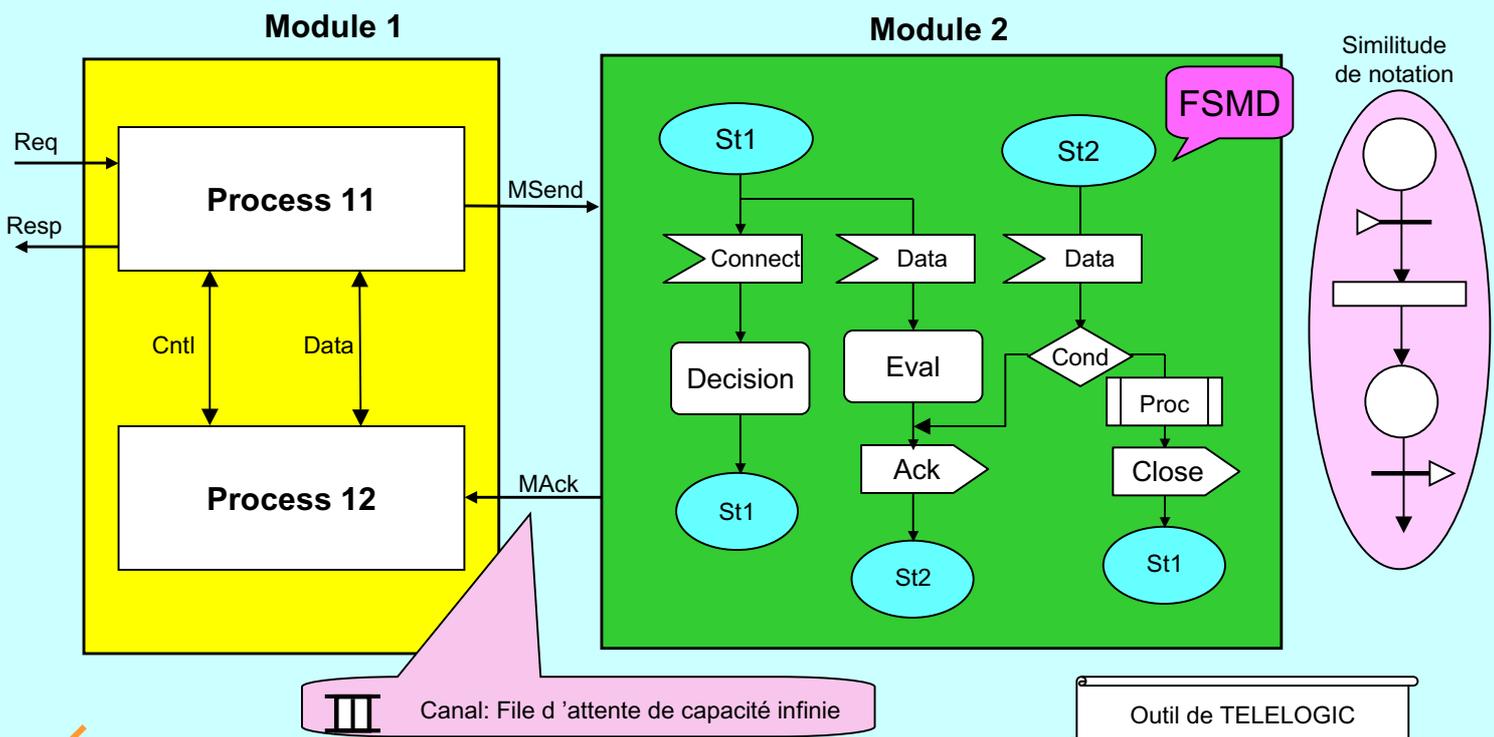
critère d'arrêt décomposition

Spécification du comportement (états discrets)

- Automate à états finis (FSM)
- GRAFCET
- StateChart
- Réseau de Petri

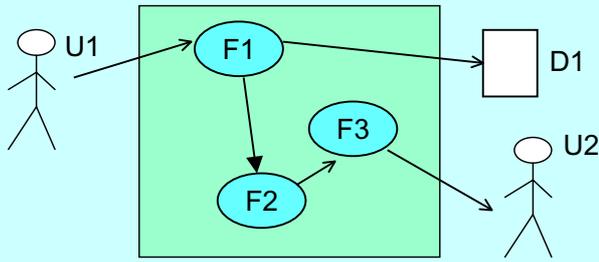


Specification and Description Language pour les systèmes de Communication

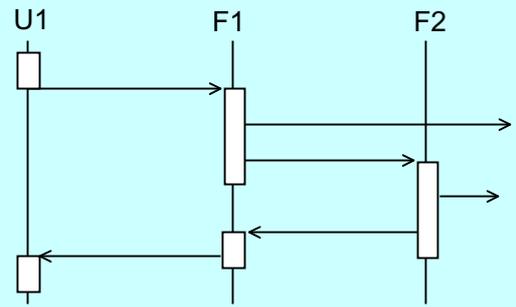


Specification: Approche Orientée Objet

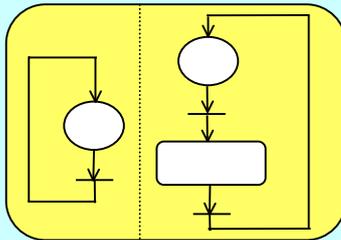
Cas d'utilisation (Use cases)



Sequence Charts (scénarios)



Modes - Behaviors

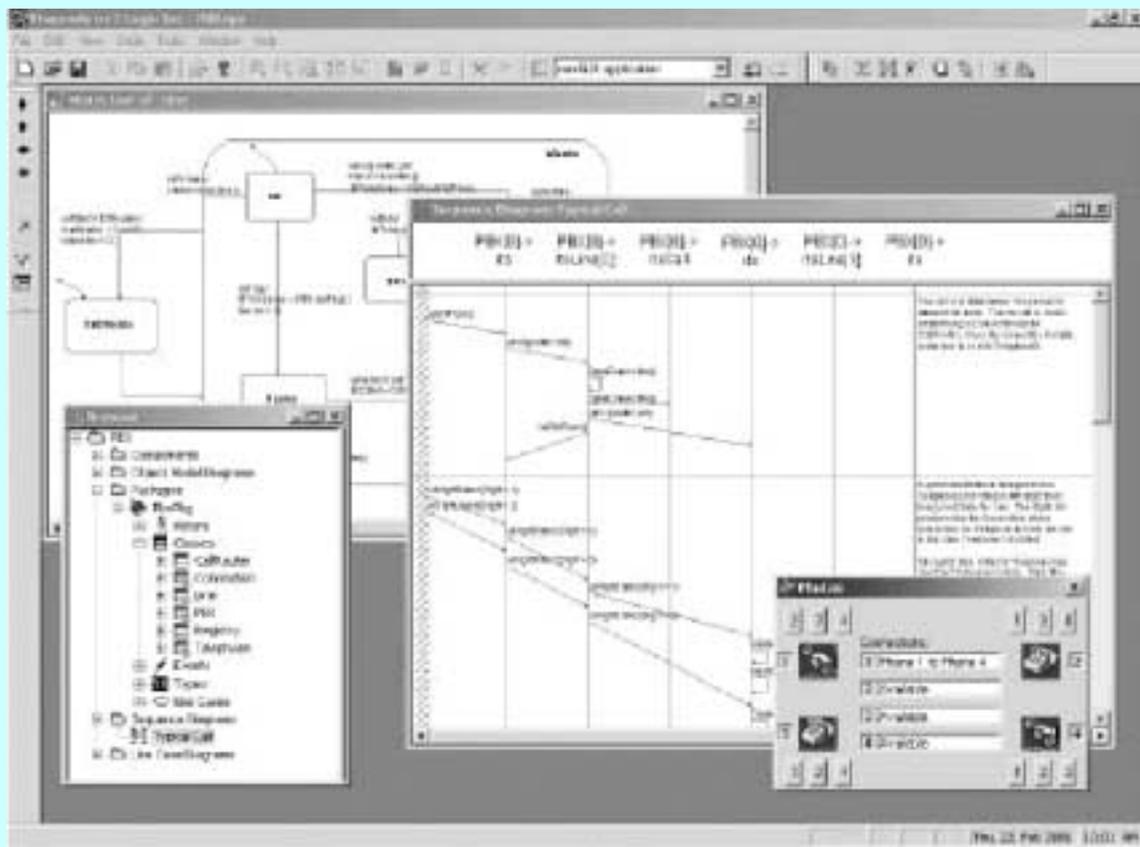


Contraintes

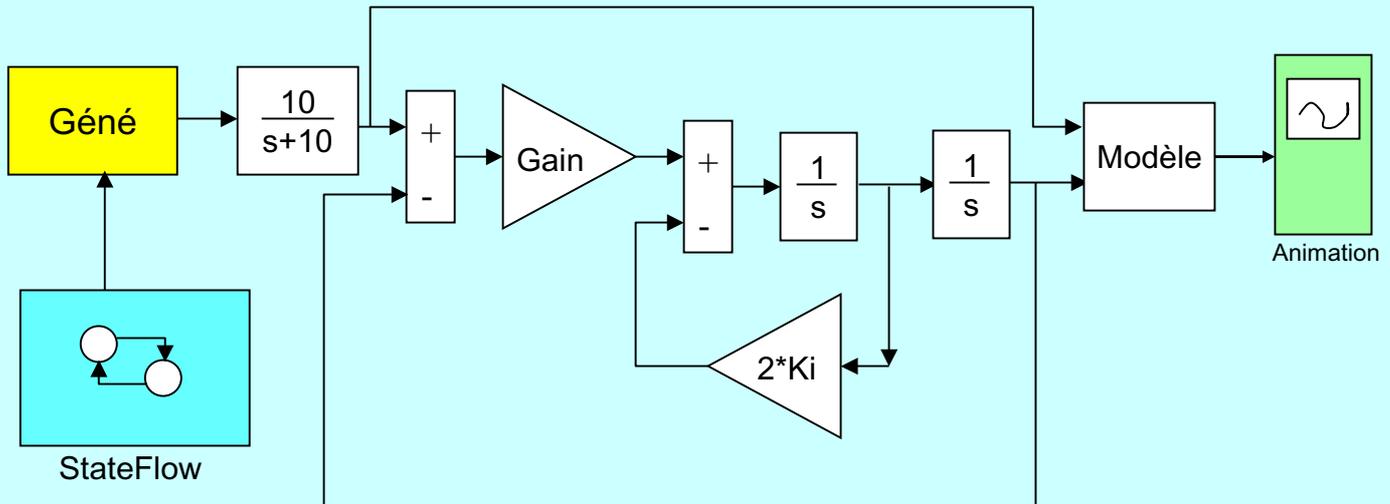
Requirement	Target	Measure
Environment		
Safety		
History		
Reliability		

Source: **Artisan** "real-time perspective"

Rhapsody (Ilogix)

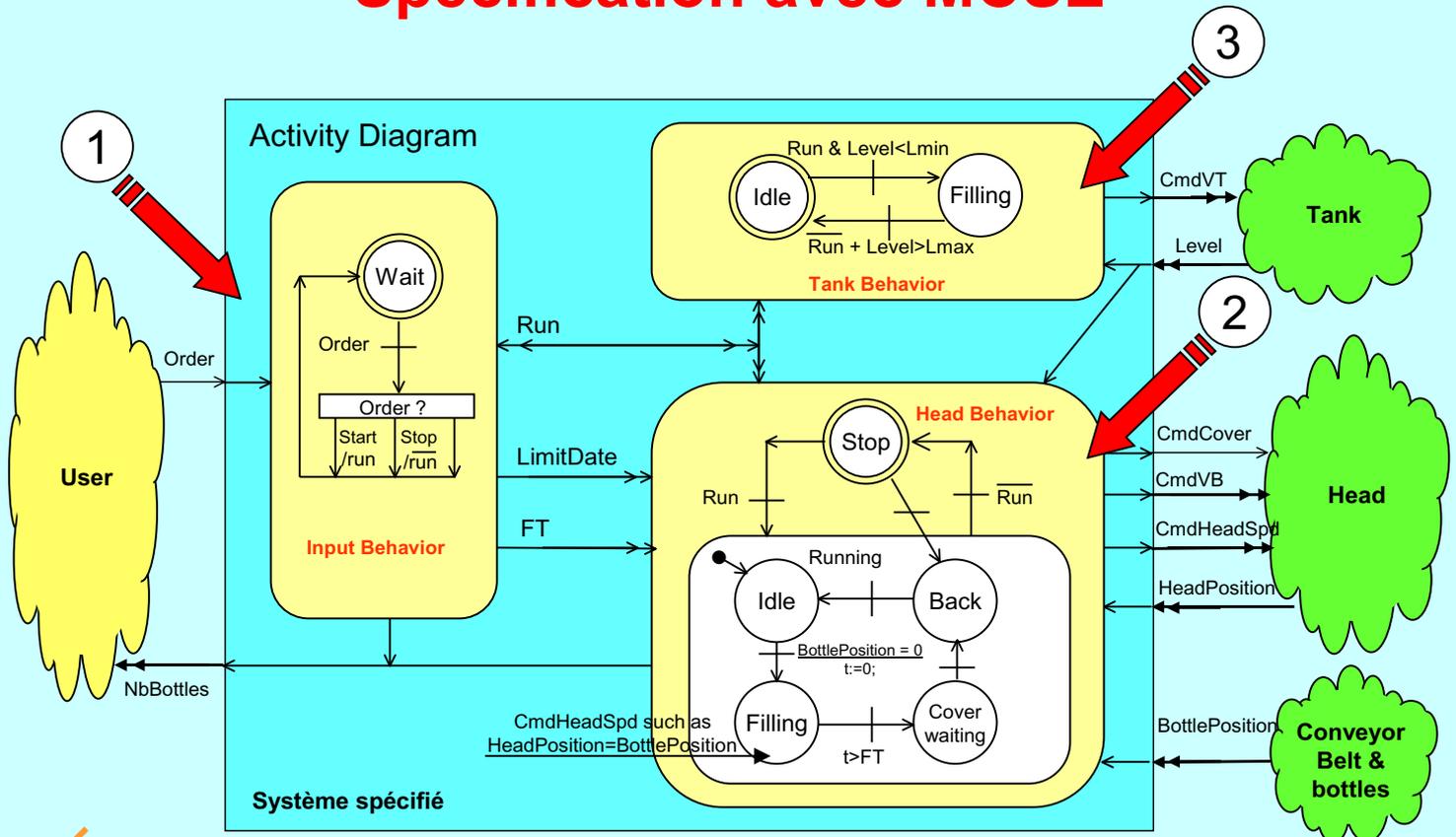


Matlab-Simulink et Matrix_x



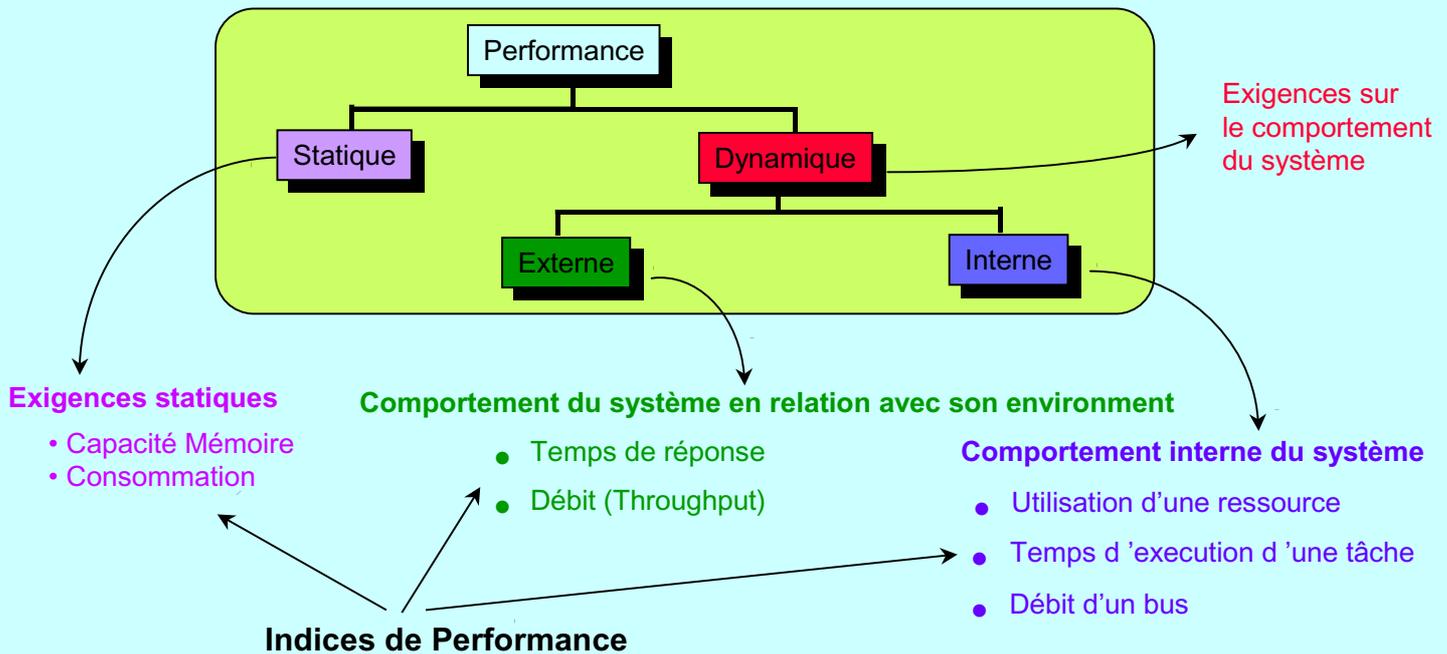
- Modélisation de systèmes dynamiques sous forme de schémas-blocs
- Bibliothèque riche de blocs de base pré-programmés
- Méthodes d'intégration, simulation, linéarisation, etc.
- Génération de code C

Spécification avec MCSE



Spécification des performances

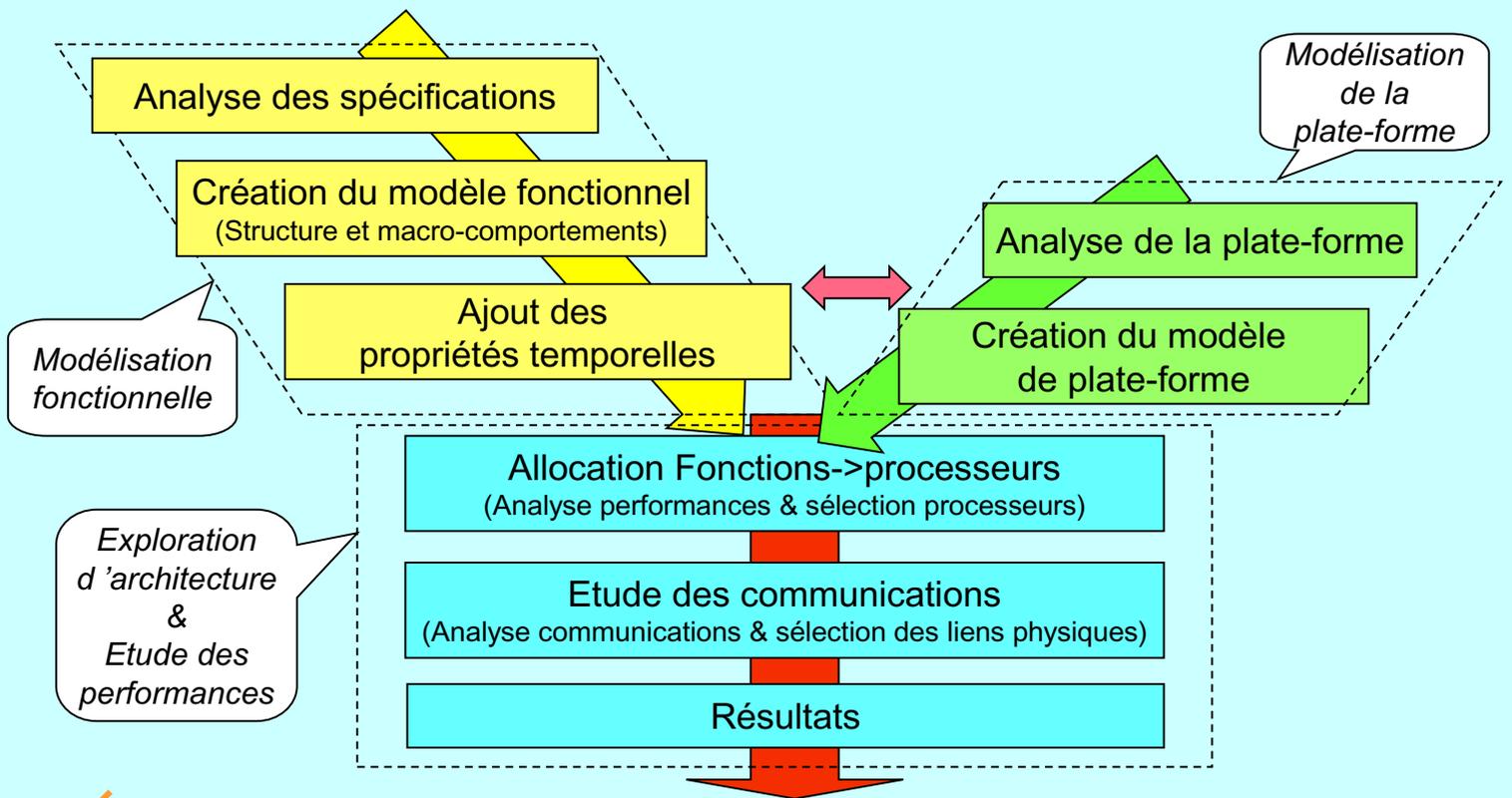
Quantification des propriétés d'un système selon des critères d'observation



Plan

- La conception système: c'est quoi
- Les bases de la modélisation et de la conception
- Spécification d'un système
- ➔ **• Conception fonctionnelle**
- Conception architecturale
- Un outil comme support
- Conclusions

Méthode globale de conception

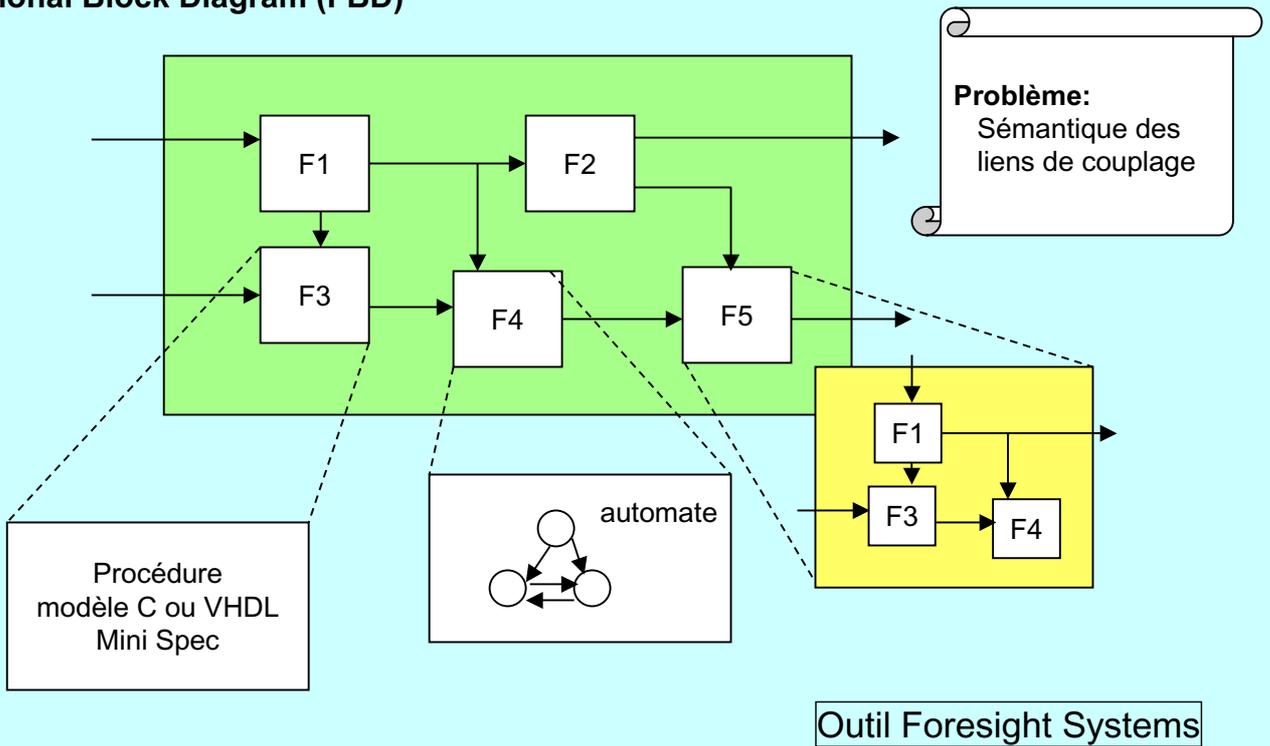


Les concepts pour la modélisation fonctionnelle

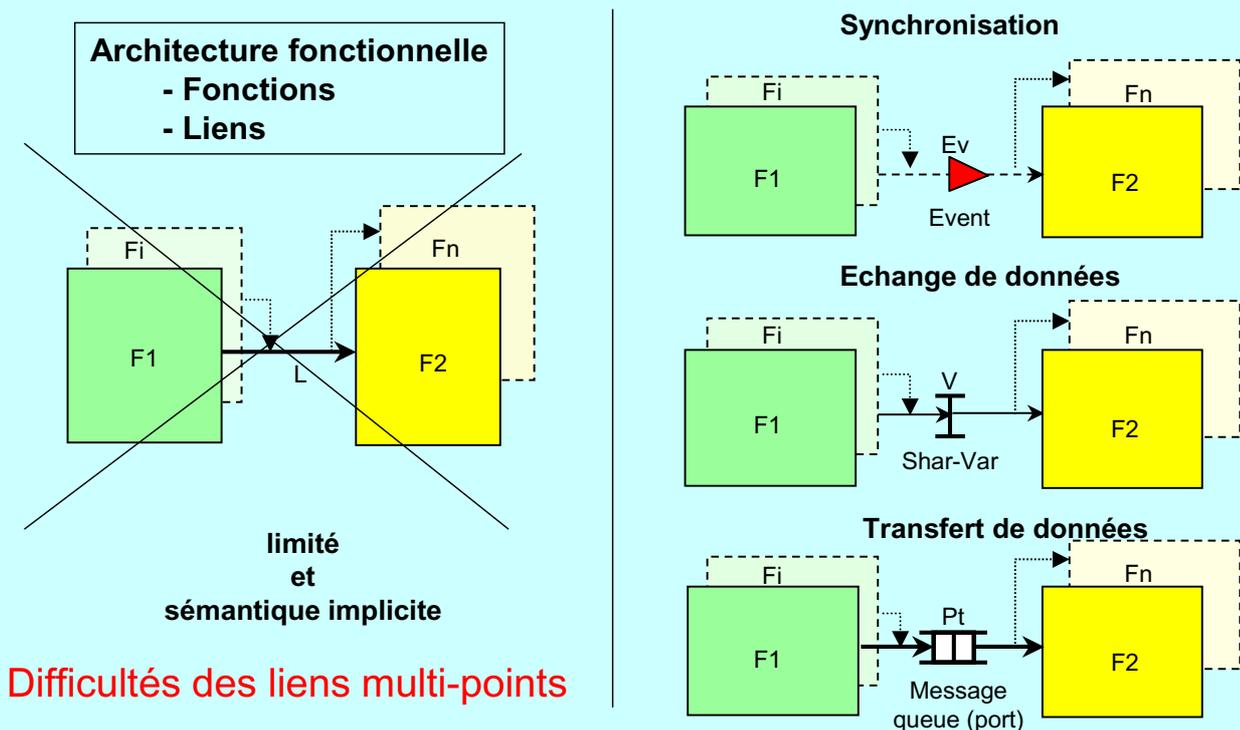
- 2 points de vue complémentaires:
 - Organisation => structure
 - rôle => comportement
- Structure
 - constituants: signification et leurs propriétés
 - interconnexions: nature et règles
 - => *modèle graphique et hiérarchique*
- Comportement
 - activités, actions opérations, états, transactions, conditions
 - règles de construction: séquence, répétition, alternance, parallélisme, conditions d'évolution
- => Difficultés:
 - Prévoir le comportement global dynamique
 - Modèle indépendant de la technologie

Exemple de modèle fonctionnel

Functional Block Diagram (FBD)

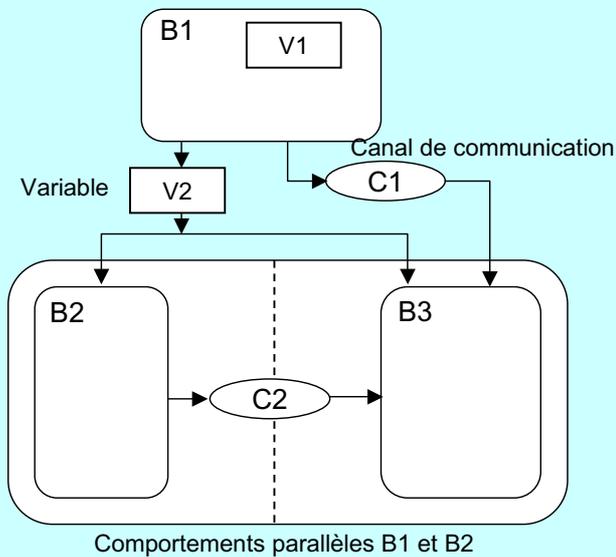


Modèle fonctionnel de MCSE (Technologie-independant)

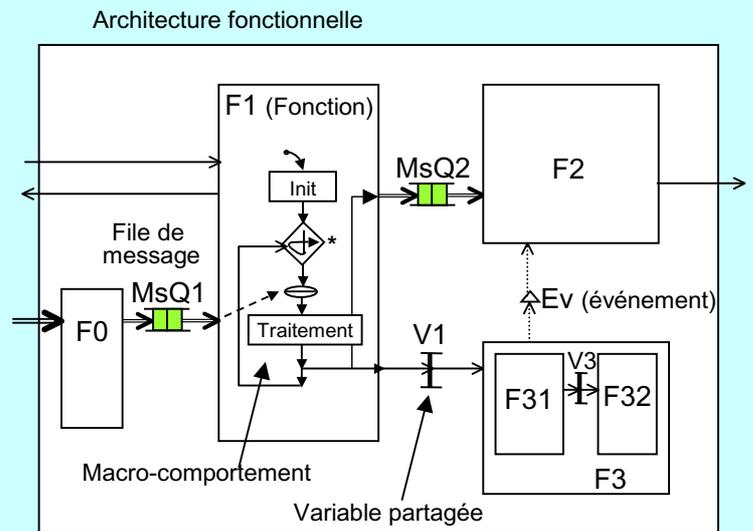


Exemples de modèles fonctionnels

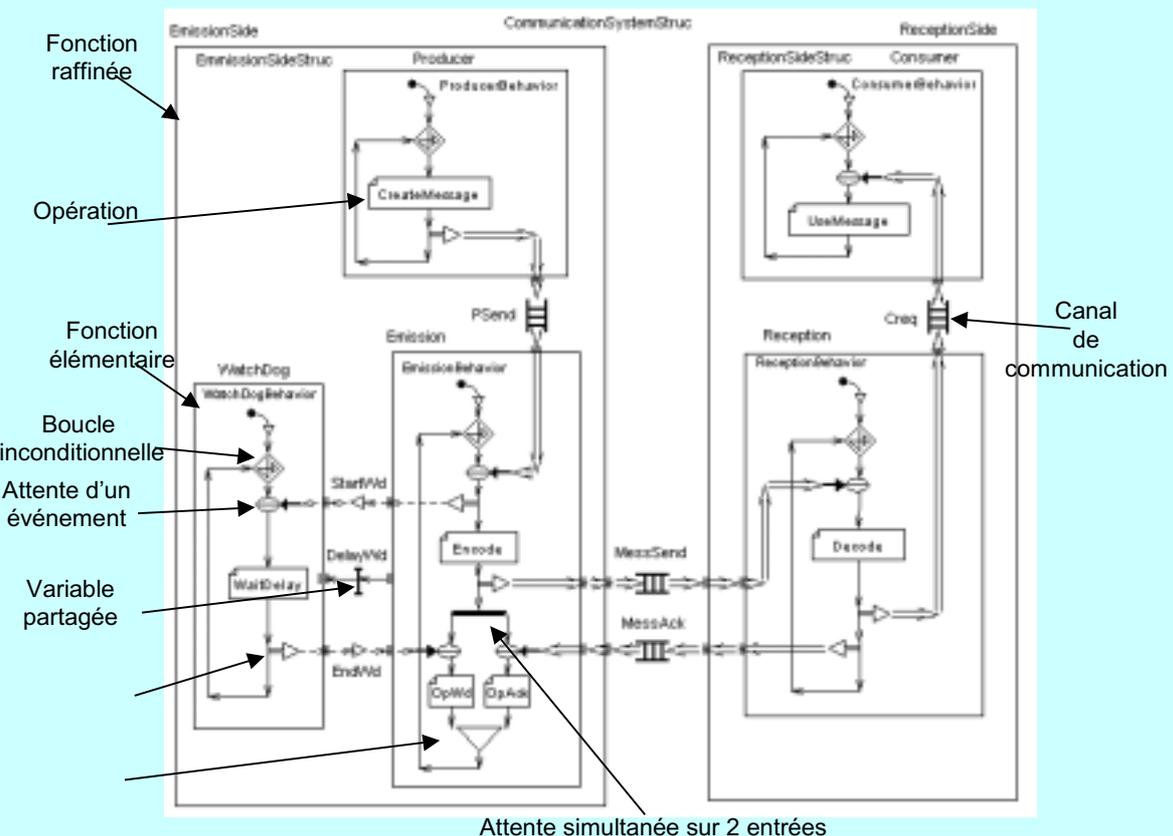
a) Modèle SpecChart



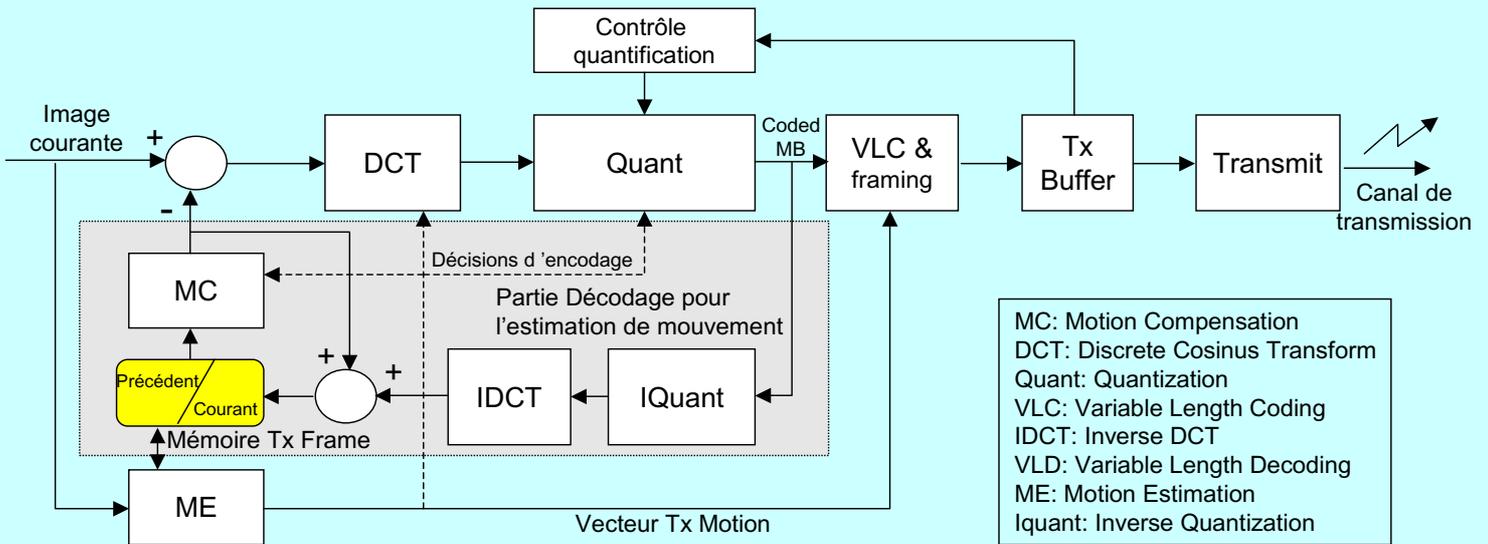
b) Modèle MCSE



Modèle MCSE: Structure & comportement

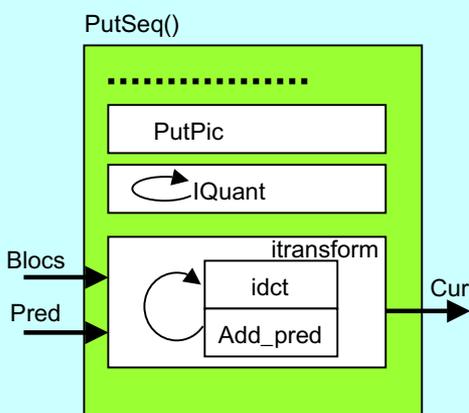


Exemple d'illustration: Codeur MPEG

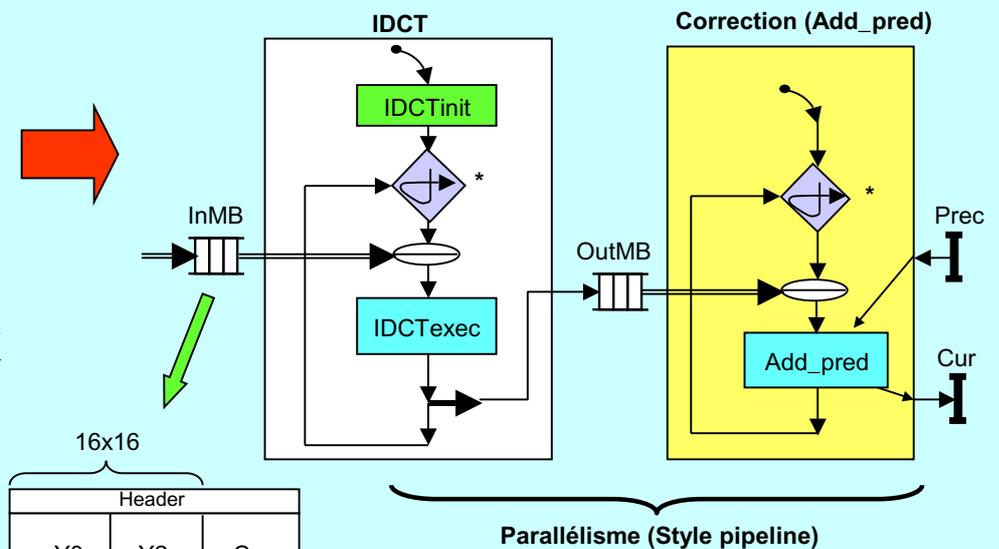


Etape 2 - Création du modèle fonctionnel

A) Description procédurale



B) Modèle fonctionnel



C) Format du MacroBloc

16x16

Header		
Y0	Y2	Cr
Y1	Y3	Cb

Exemple: partie codage et estimation

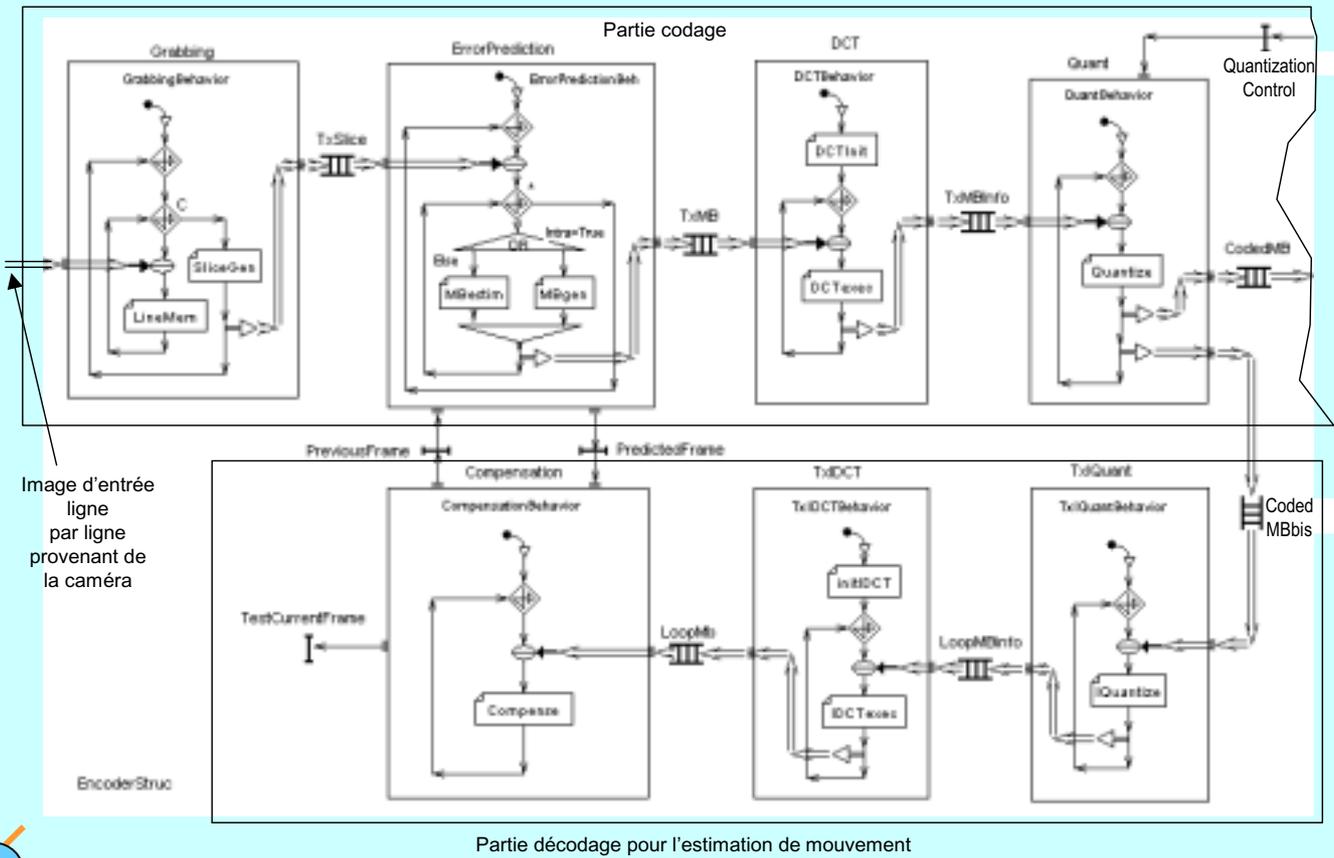
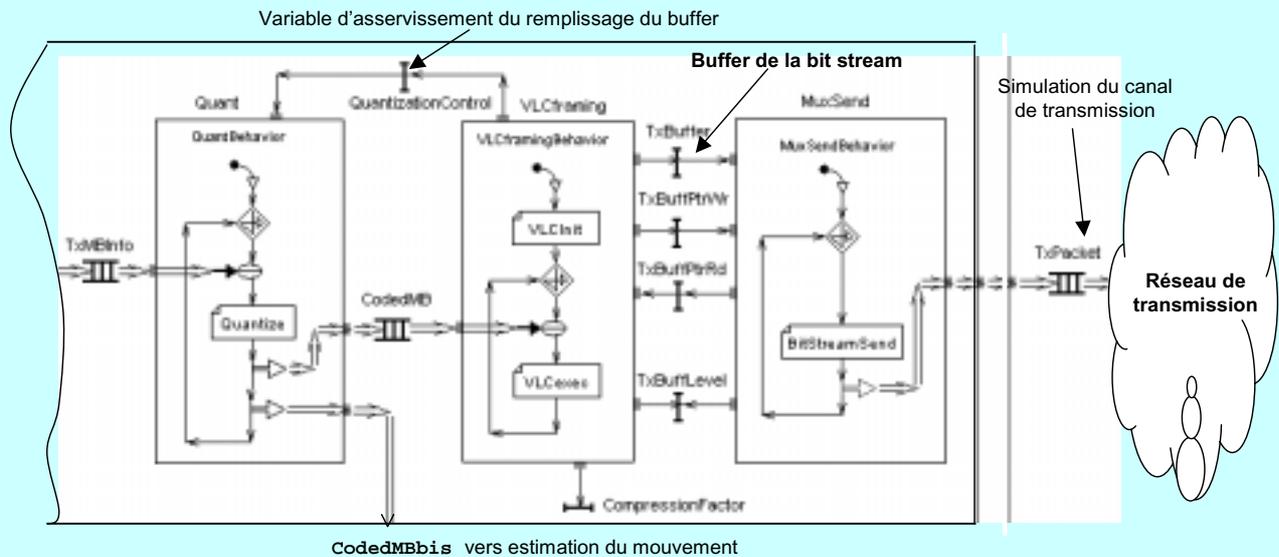


Image d'entrée ligne par ligne provenant de la caméra

Exemple: partie compression et transmission



Variable d'asservissement du remplissage du buffer

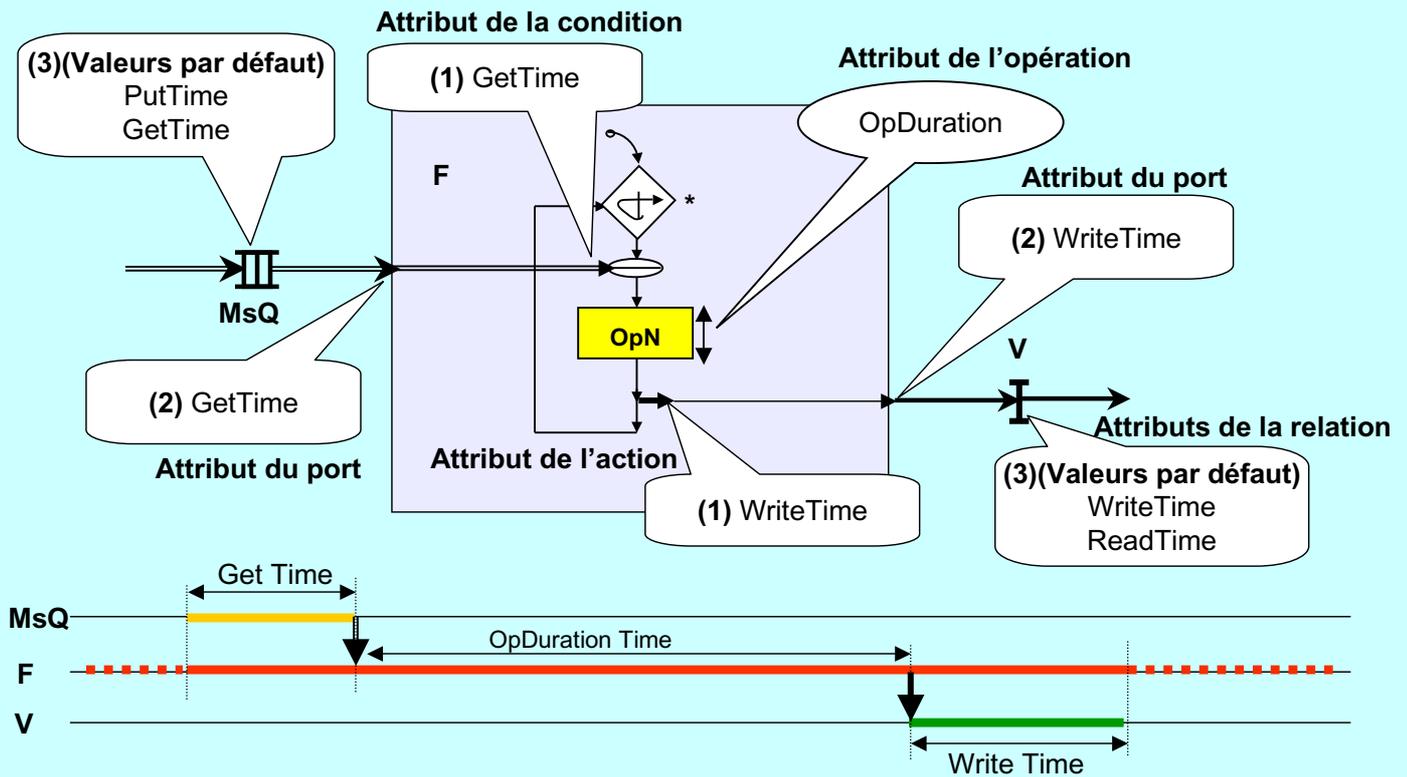
Buffer de la bit stream

Simulation du canal de transmission

Réseau de transmission

CodedMBbis vers estimation du mouvement

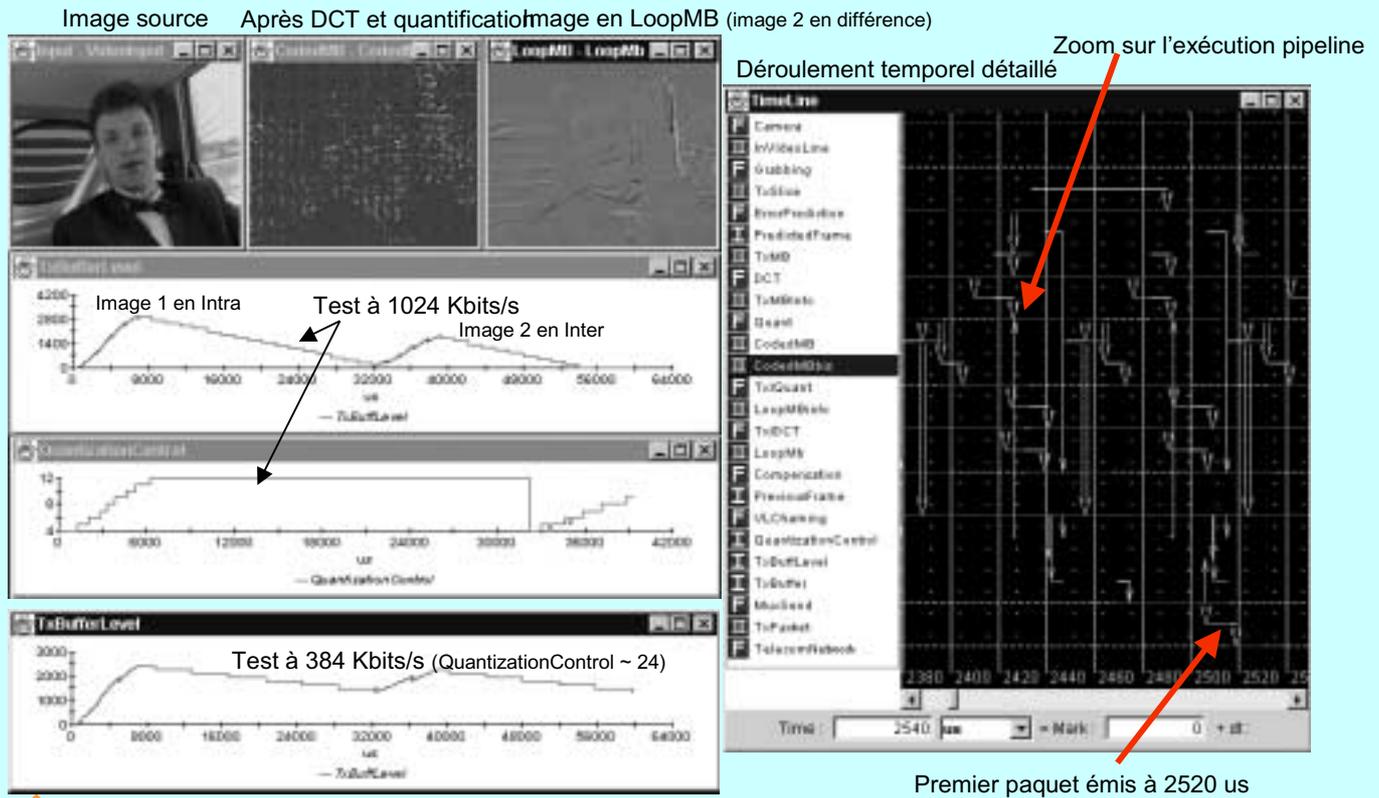
Paramétrage temporel du modèle



Types d'attributs temporels

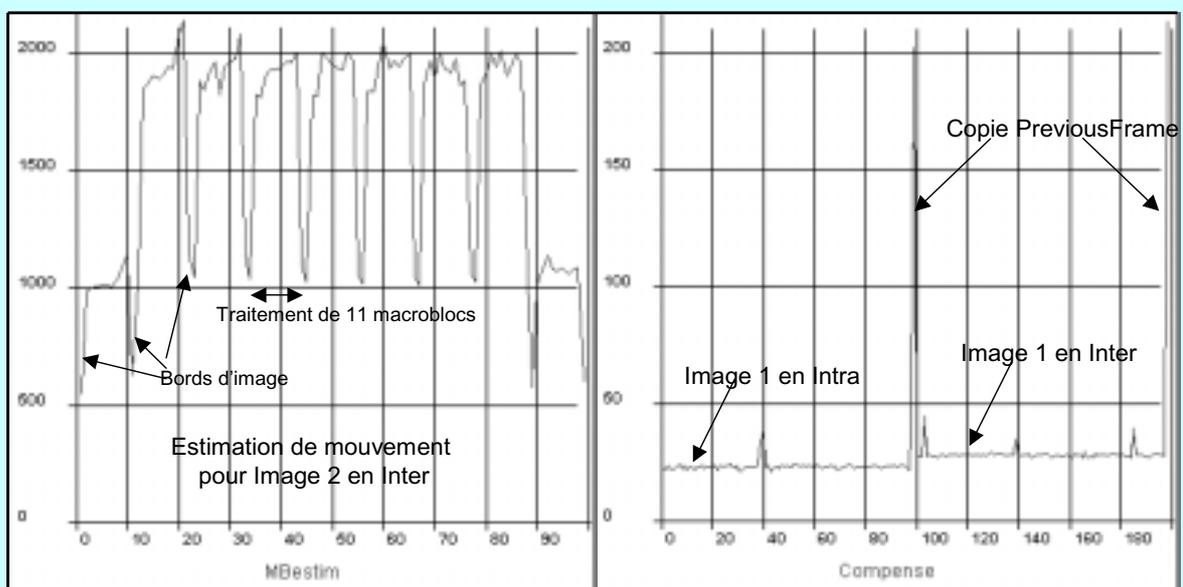
- Constante, Min/Nom/Max
- Toute expression mathématique (exprimable en C/C++)
- Valeurs en ...Sec ou en cycles ($t = \text{NbCy} \times \text{CyclePeriod}$)
- Mots clés prédéfinis pour les communications:
 - DATASIZE,
 - FCTCOMPUTETIME (méthode définie par l'utilisateur)
- Temps d'exécution pour chaque Opération
 - Prédéfini, Profilé, fonctions aléatoires
 - dépendant des données => Evalué durant l'exécution de l'algorithme
- Utilisation de paramètres générique dans toute expression
- Surcharge d'attributs de temps

Vérification par simulation: exemple de résultats



Estimation des temps d'exécution

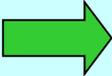
- Evaluation sur un CPU: modèle de CPU et/ou ISS
- Evaluation en matériel: synthèse architecturale
- Profiling des algorithmes



Conception fonctionnelle : Méthode

- Approche indépendante de la technologie
- Raffinement et Dédution à partir des spécifications et pas assemblage de composants
- Arrêt du raffinement lorsque les fonctions se décrivent par un algorithme séquentiel
- Utiliser des modèles génériques ou design patterns comme guide
- Vérifier et valider la conception par simulation (si possible)
- Commencer à calibrer le modèle en le « timant » au mieux

Plan

- La conception système: c'est quoi
- Les bases de la modélisation et de la conception
- Spécification d'un système
- Conception fonctionnelle
-  **Conception architecturale**
- Un outil comme support
- Conclusions

Problèmes de la conception architecturale

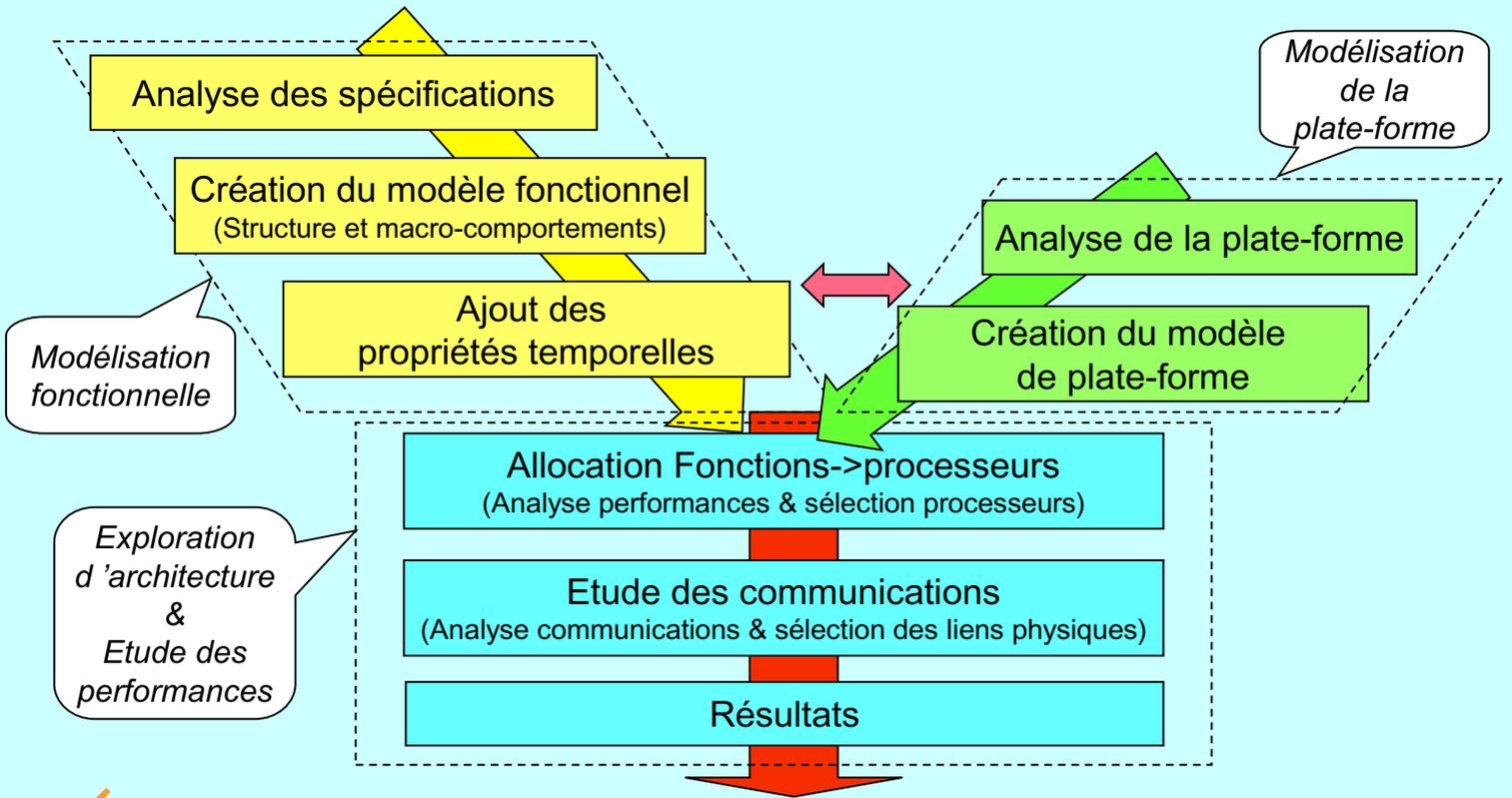
- Types d'architectures
 - Architecture imposée ou à déterminer
 - Architecture simple (CPU + Asic) ou architecture hétérogène distribuée, maître/esclave ou multi-maître
- Niveaux de granularité
 - Tâche ou process ou fonction
 - Bloc
 - Instruction
- Difficultés du problème
 - Partitionnement Hw/Sw
 - Sélection d'architecture(s)
 - Sélection de composants
 - Estimation des propriétés (What-If-Analysis)

==> Partitionnement Interactif plutôt qu'automatique

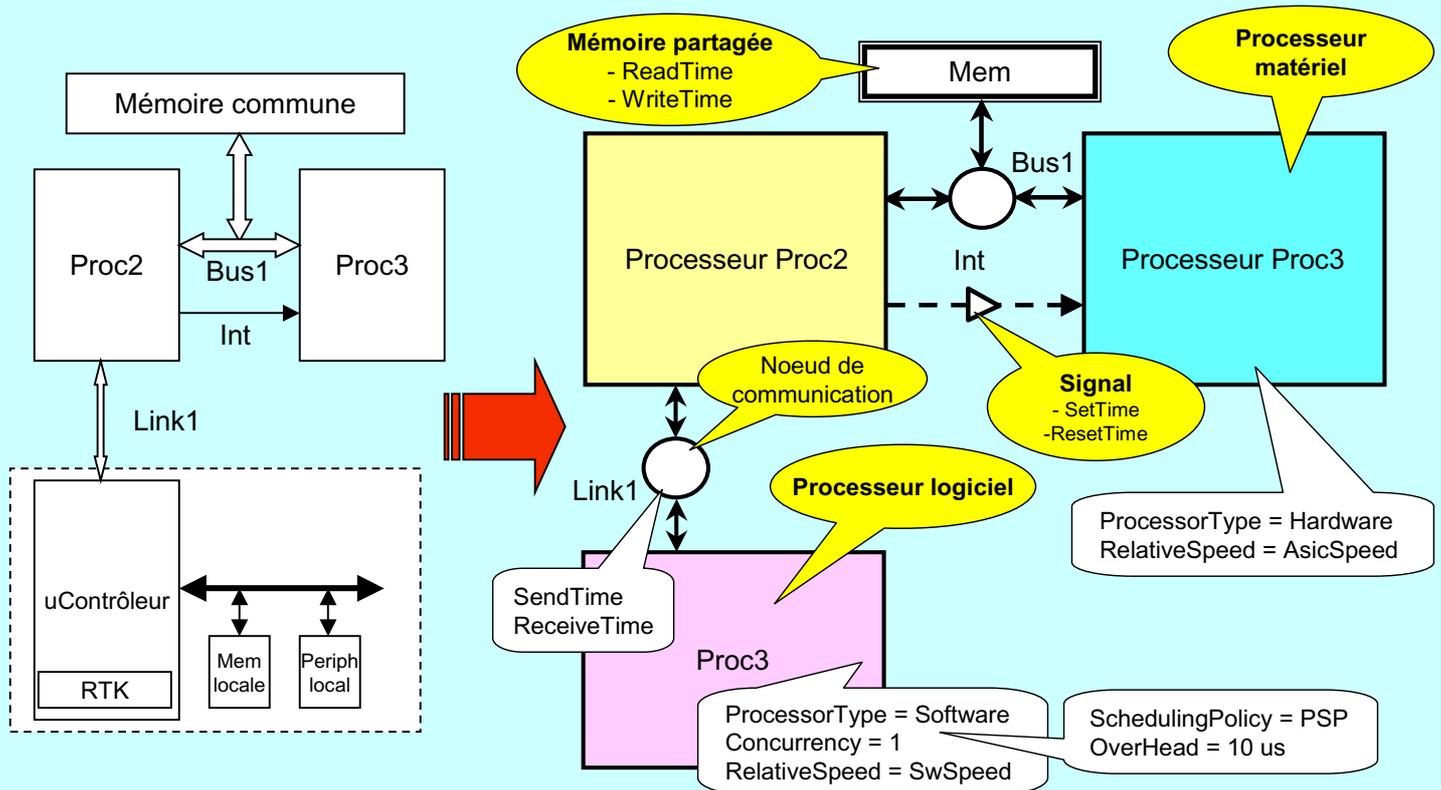
Problèmes de l'exploration d'architectures

- Estimation en matériel
 - synthèse architecturale de la fonction (algorithme)
 - résultat: temps d'exécution (Nb cycles), surface, consommation
- Estimation en logiciel
 - calcul des performances par analyse du source (jeu d'instructions)
 - résultat: temps d'exécution max, taille code, taille données (cache?)
- Estimation en mémoire
 - détermination des tailles mémoire, hiérarchie des mémoires
 - résultat: architecture mémoire, capacités, bus d'interconnexion
- Estimation des communications
 - sélection des protocoles de communication et performances
 - résultat: mécanismes de communication, latence, débit
- **==> Nécessité d'un bon modèle d'estimation pour l'exploration**

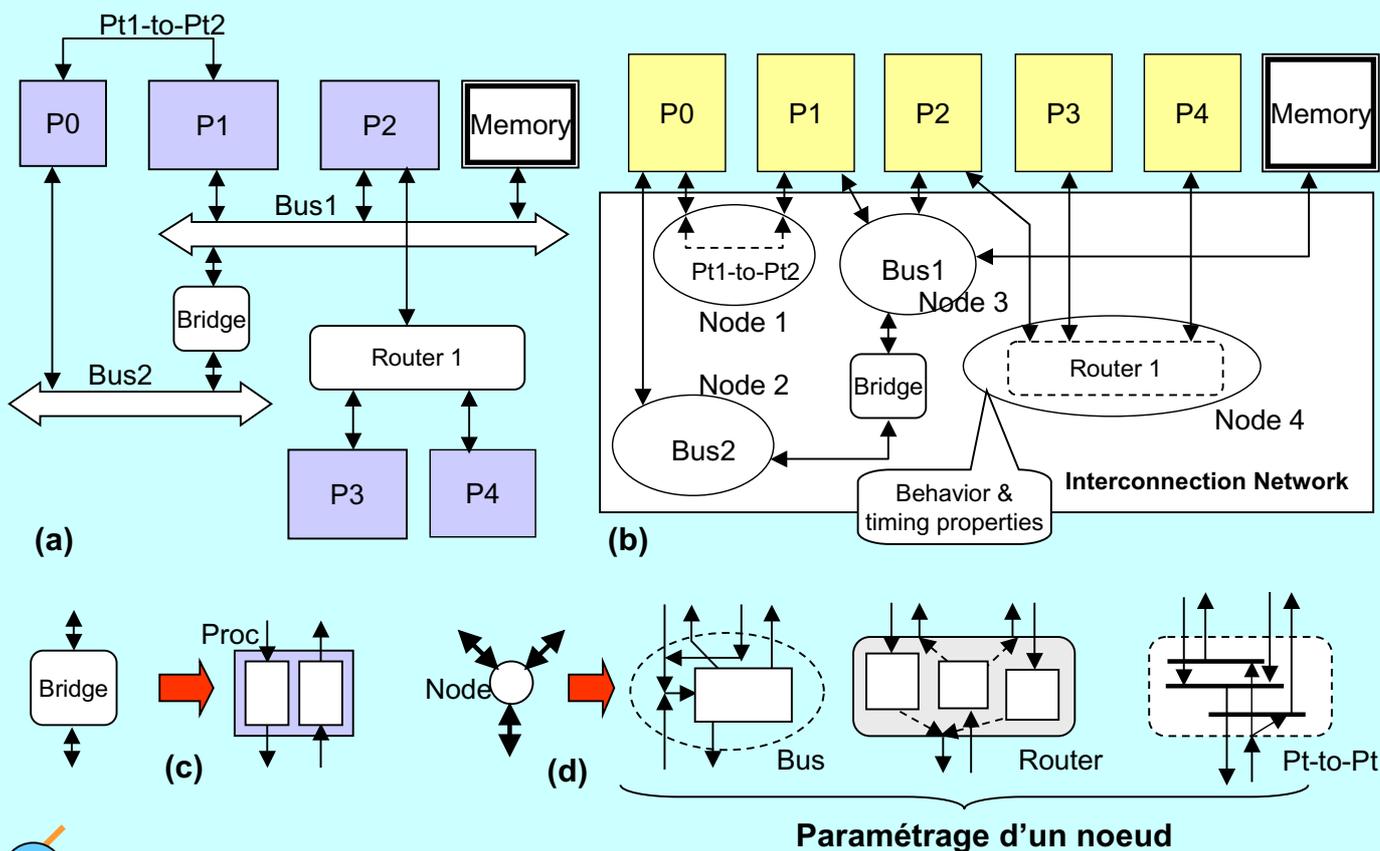
Méthode pour la conception architecturale



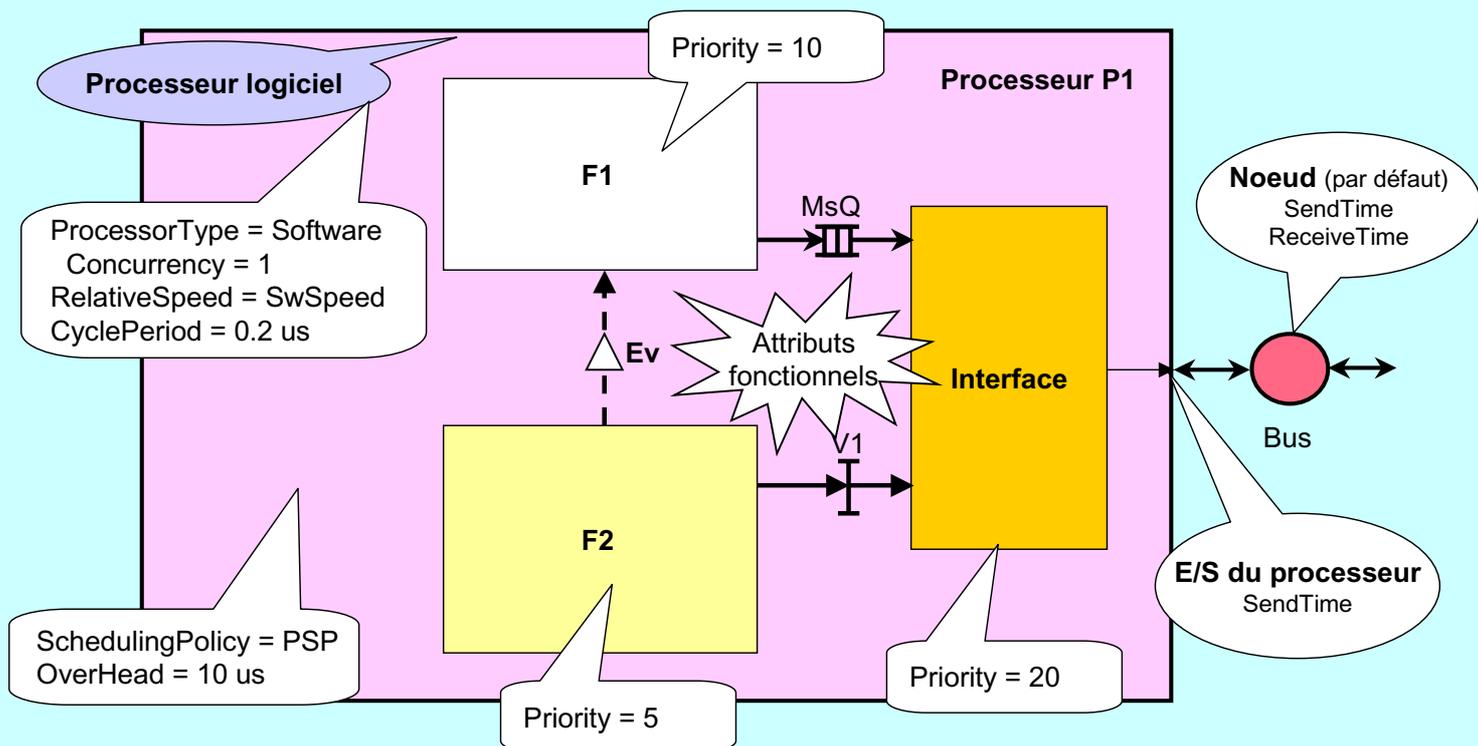
Modélisation d'une plate-forme



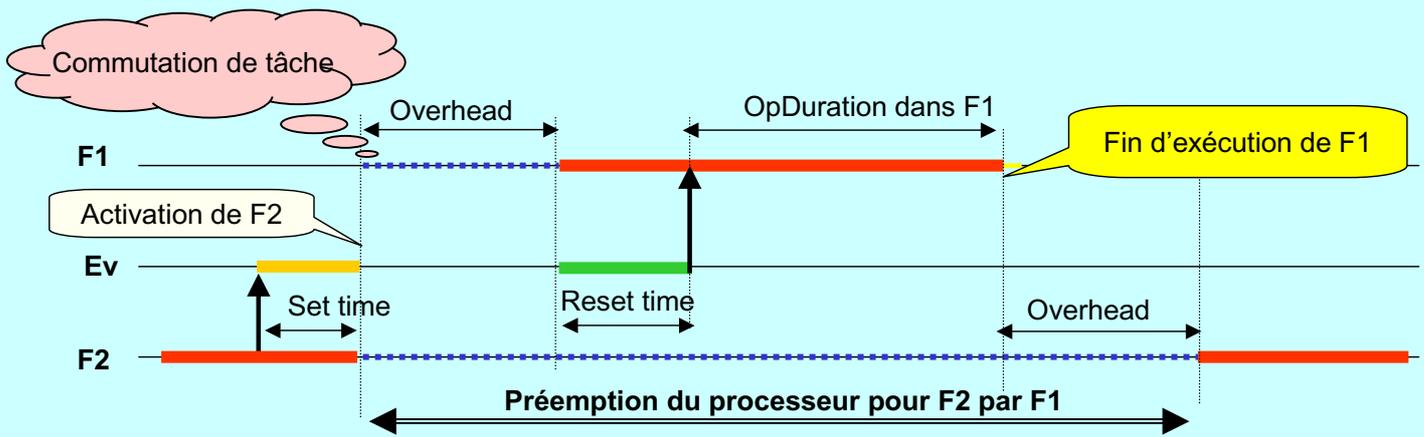
Modèle d'interconnexion physique



Attributs pour un processeur logiciel

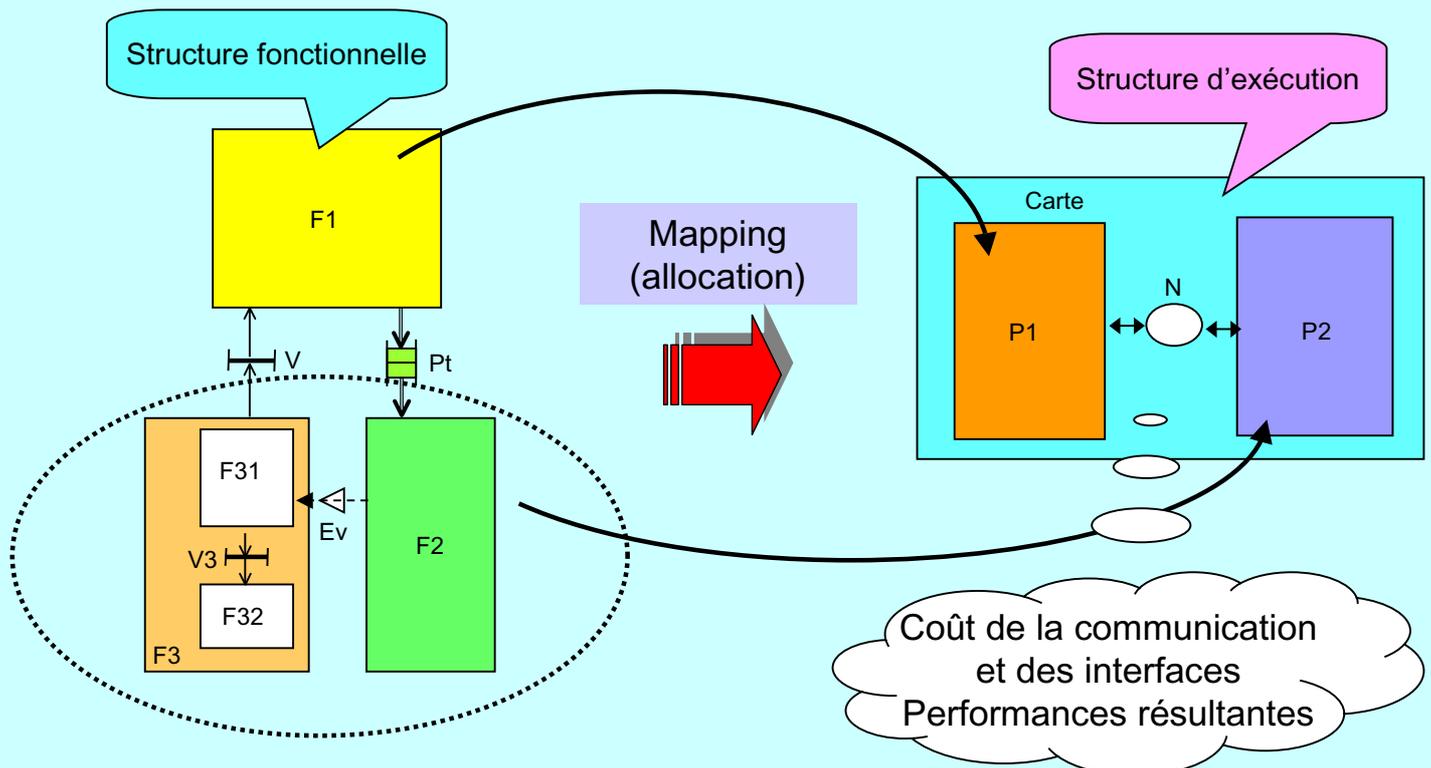


Comportement pour un processeur logiciel



- $T_{exec} \text{ sur Sw/Hw} = OpDuration [* CyclePeriod] / RelativeSpeed$
- Attributs fonctionnels surchargés par ceux du processeur

Etape 3 - Partitionnement et Allocation



Résultats de performances attendus

- Précision des grandeurs et temps de réactivité
- Diagramme temporel détaillé (timeline)
- Vérification des échéances temporelles
- Taux d'occupation des ressources et fonctions (tâches)
- Taux d'utilisation des liens d'échange physiques
 - Goulot d'étranglement
 - Effet de la latence des communications
- Profiling des algorithmes (optimisation et alloc de perf)

Performances globales: sans lien physique

60 ms: Traitement des 2 premières images

Fct/Pro	Active	Idle	CPU preempted	Res. waiting
P PhyBoard	88.7 %	11.3 %	0.0 %	0.0 %
P MotionProc	34.4 %	65.6 %	0.0 %	0.0 %
P CodingProc	79.4 %	20.6 %	0.0 %	0.0 %
P ControlProc	27.9 %	72.1 %	0.0 %	0.0 %
F Grabbing	5.2 %	94.8 %	0.0 %	0.0 %
F ErrorPrediction	19.0 %	81.0 %	0.0 %	57.7 %
F Compensation	13.5 %	86.5 %	0.0 %	0.0 %
F DCT	22.1 %	22.3 %	40.3 %	15.4 %
F Quant	12.5 %	23.2 %	56.2 %	7.7 %
F TxQuant	11.9 %	37.9 %	30.5 %	19.3 %
F TxIDCT	21.8 %	30.2 %	47.6 %	0.0 %
F VLCframing	22.5 %	74.6 %	2.8 %	0.0 %

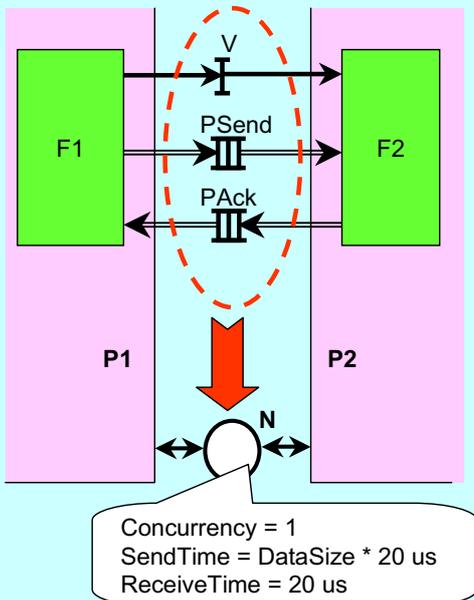
Relation Element	Used	Read	Write	Idle
III TxSlice	3.4 %	1.7 %	1.7 %	96.6 %
I PredictedFrame	4.0 %	2.6 %	1.3 %	96.0 %
I PreviousFrame	1.4 %	0.2 %	1.3 %	98.6 %
III CodedMB	5.5 %	2.8 %	2.6 %	94.5 %
III LoopMb	5.3 %	2.6 %	2.6 %	94.7 %
III TxMB	5.3 %	2.6 %	2.6 %	94.7 %
III TxMBInfo	5.3 %	2.6 %	2.6 %	94.7 %

RelativeSpeed = 1

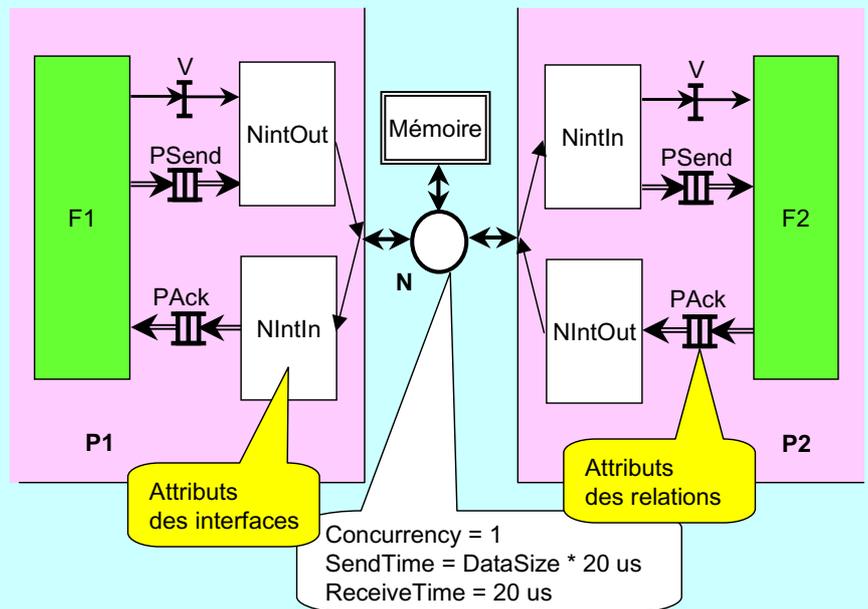
Temps codage image: 25 ms, CodingProc: 79,4% ou 39,7% (speed x 2)

Modélisation pour les liens physiques

A) Partage du noeud N sans interfaces



B) Partage du noeud N avec interfaces



Performances avec les liens physiques

Statistics of TimeLineWithBus
Start: 0 us End: 60000 us dt: 60000 us

FcbPro	Active	Idle	CPU preempted	Res. waiting
P PhyBoard	100.0 %	0.0 %	0.0 %	0.0 %
P MotionProc	45.5 %	54.5 %	0.0 %	0.0 %
MotionProc/BusInInterface	8.6 %	91.4 %	0.0 %	0.0 %
MotionProc/BusOutInterface	8.5 %	90.9 %	0.0 %	0.8 %
P CodingProc	99.3 %	0.7 %	0.0 %	0.0 %
CodingProc/BusOutInterface	10.0 %	60.5 %	29.5 %	0.0 %
CodingProc/BusInInterface	3.5 %	74.2 %	21.0 %	1.2 %
P ControlProc	46.0 %	54.0 %	0.0 %	0.0 %
ControlProc/BusInInterface	4.2 %	91.6 %	0.0 %	0.0 %
ControlProc/BusOutInterface	5.1 %	87.8 %	0.0 %	0.0 %
M SharedMem	14.3 %	85.7 %	0.0 %	0.0 %

Relation Element	Used	Read	Write	Idle
Bus	14.1 %	2.5 %	11.6 %	85.9 %
I PredictedFrameInSharedMem	4.7 %	3.1 %	1.6 %	95.3 %
I PreviousFrameInSharedMem	1.6 %	1.6 %	0.0 %	98.4 %
I TxBufferInSharedMem	3.8 %	0.7 %	3.1 %	96.2 %
I PredictedFrame	9.3 %	4.6 %	4.7 %	90.7 %
I PreviousFrame	3.1 %	1.6 %	1.6 %	96.9 %

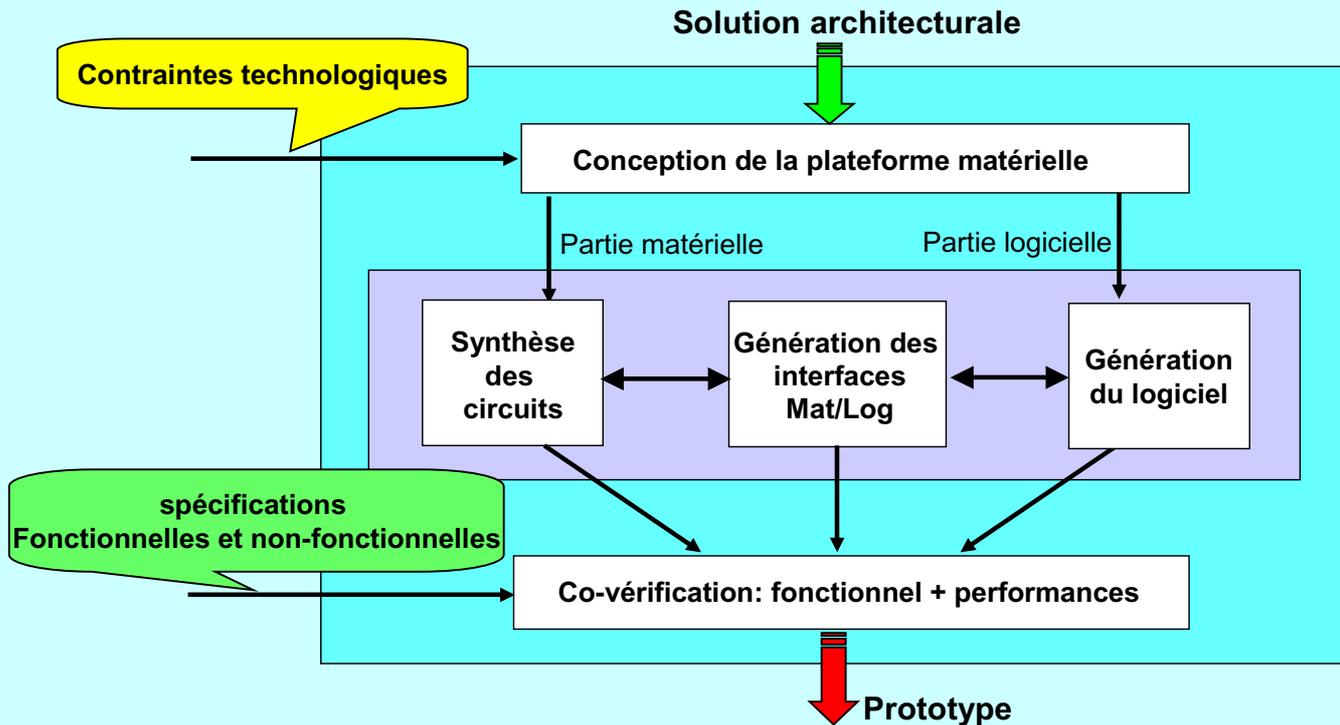
Mémoire

Bus

Interfaces pour CodingProc

Temps codage image: 32 ms, CodingProc: 99,3% ou 53,5% (speed x 2)

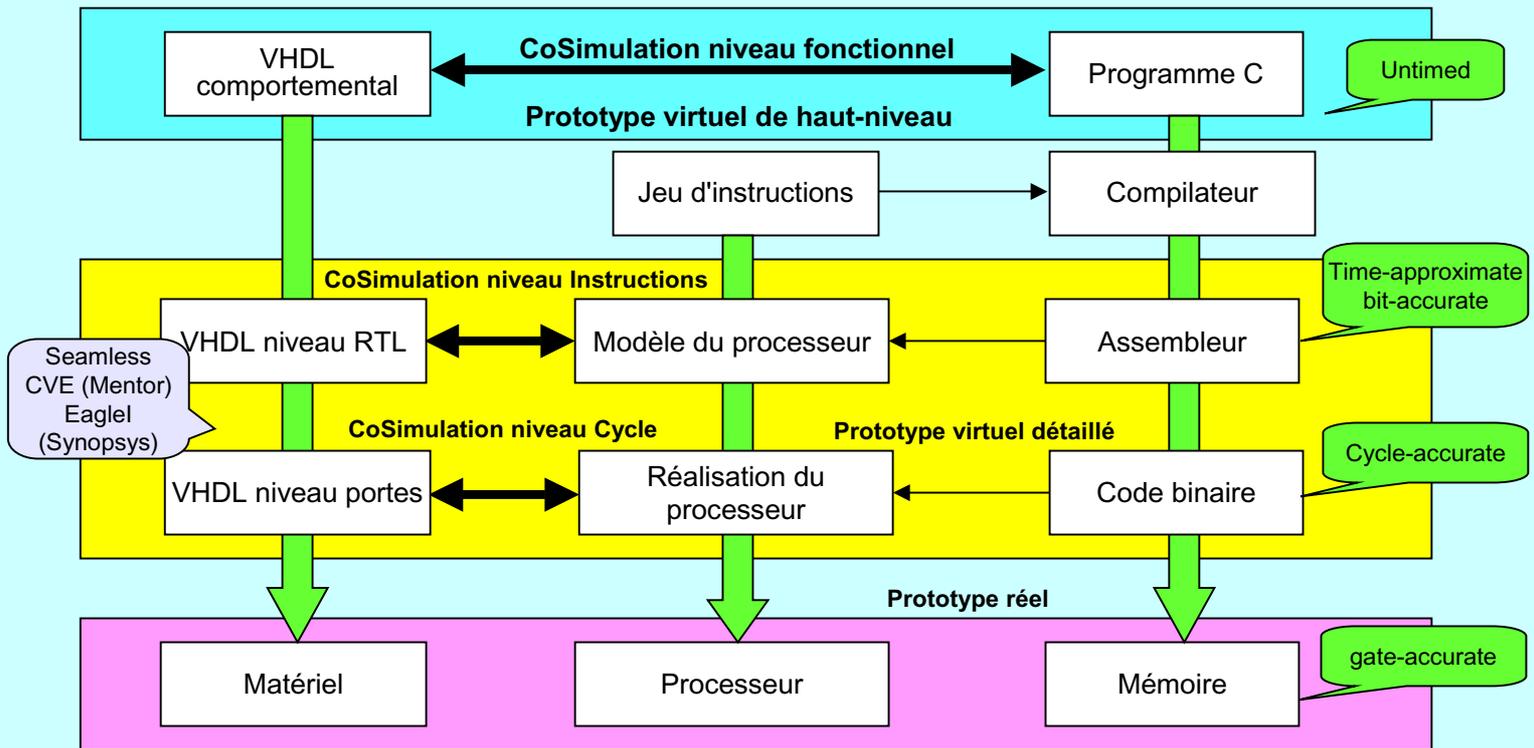
Etape 4 - Prototypage & Implémentation



Co-Vérification

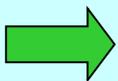
- Co-Simulation
 - Vérification de la conception fonctionnelle puis architecturale
 - Simulation du matériel et du logiciel, avant et/ou après la co-synthèse
 - Importance des modèles pour le matériel et pour le logiciel:
Compromis: précision /temps de simulation
 - Difficulté de l'estimation de performances
puis de la vérification/validation
- Prototype virtuel
 - Mono-langage ou multi-langages
 - Logiciel seul ou logiciel/matériel
- Emulation

Niveaux de CoSimulation et Prototypage virtuel

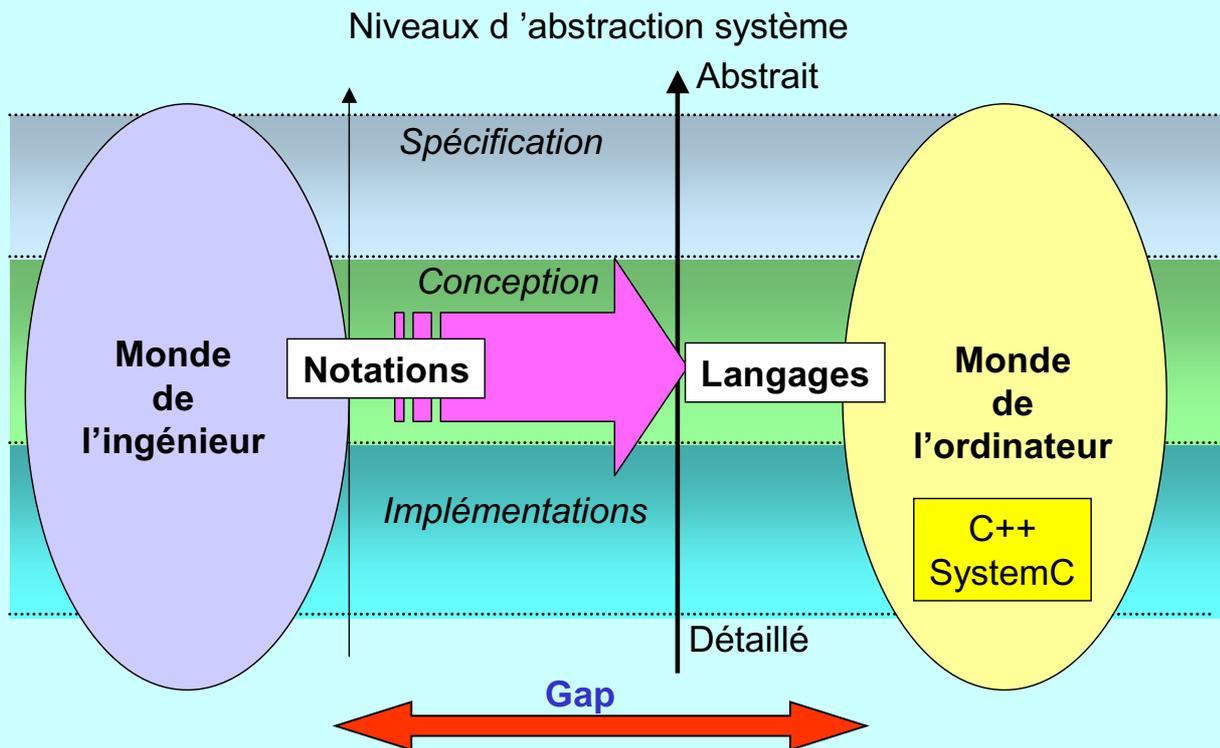


Plan

- La conception système: c'est quoi
- Les bases de la modélisation et de la conception
- Spécification d'un système
- Conception fonctionnelle
- Conception architecturale
- **Un outil comme support**
- Conclusions



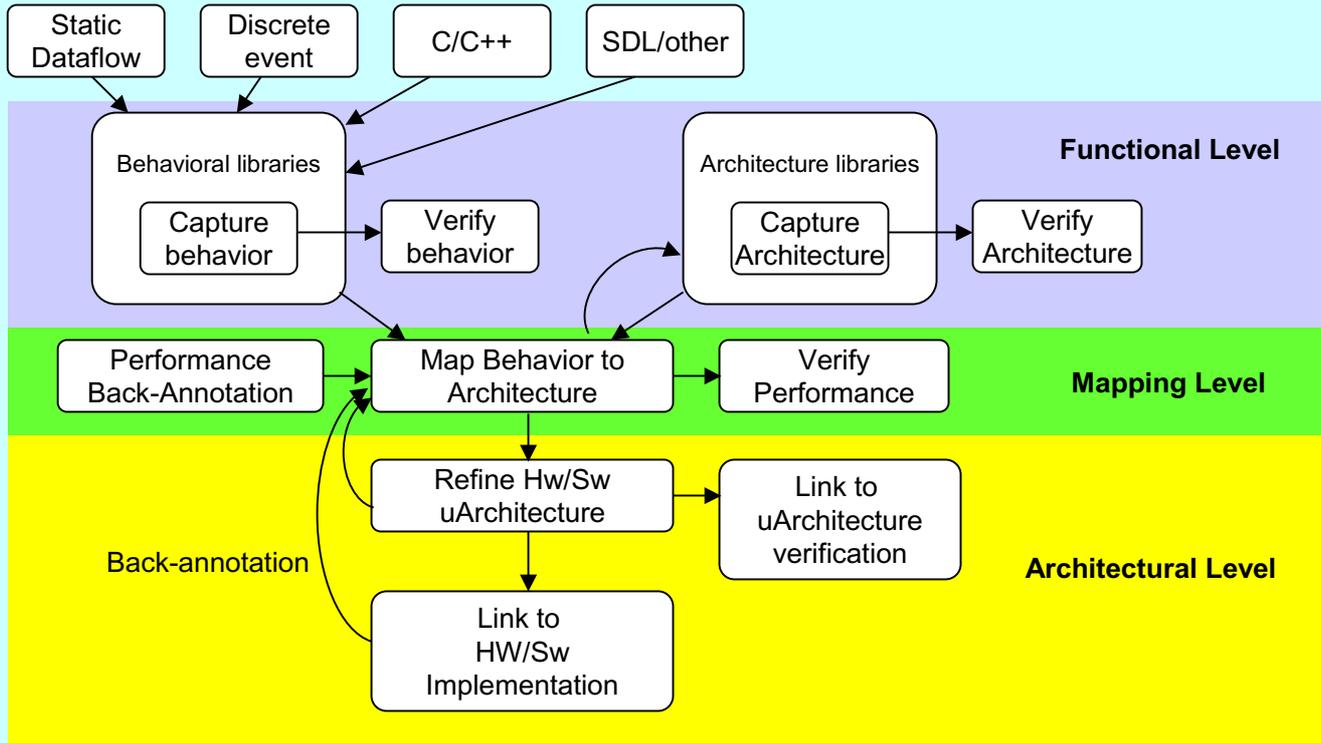
Un outil pour la conception système - Objectif



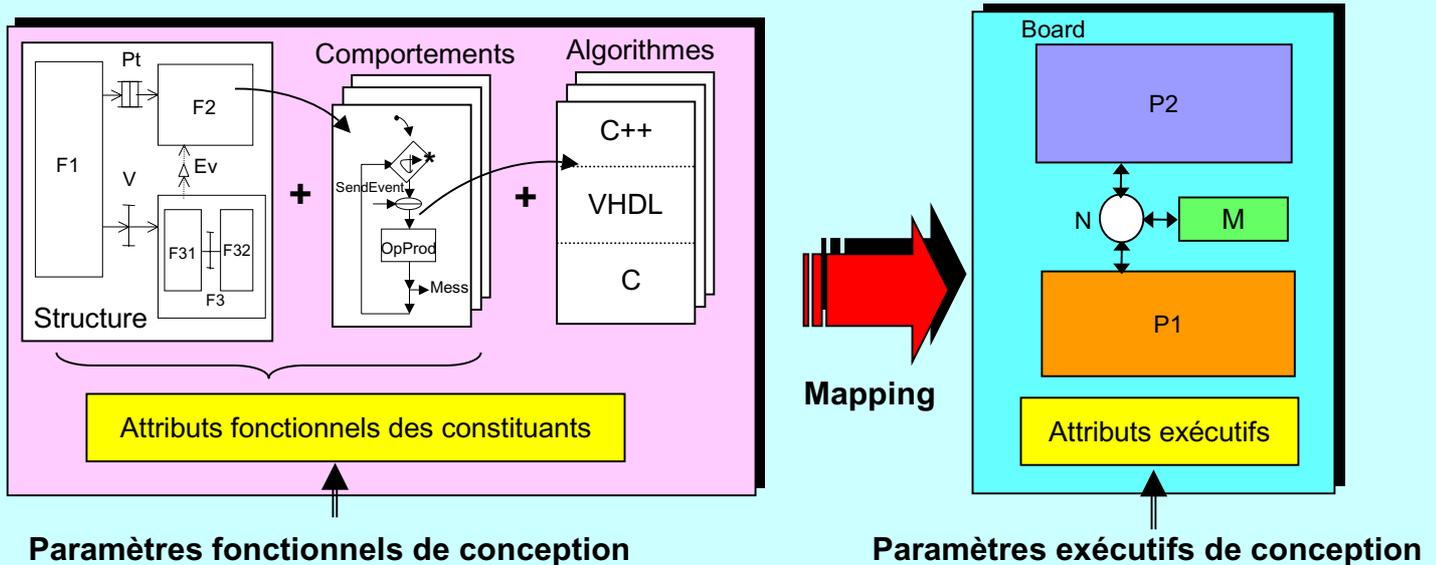
Point de vue

- Niveau du «langage»: Apte à l'expression du fonctionnel
 - Favorise les solutions de qualité et technologie-independant
 - Permet la vérification fonctionnelle et l'étude des performances
 - Facilite la génération des prototypes
- Nature du «langage»: Naturel et efficace pour les concepteurs
 - Adapté à la culture des ingénieurs: concepts et notations
 - Graphique and textuel
 - Séparation des aspects: structure, comportements, propriétés.....
 - Traduction automatique dans un langage de l'ordinateur
 - Description résultante compréhensible par les concepteurs
- Le langage n'est pas suffisant:
 - Méthodes and Méthodologie pour la qualité et l'efficacité

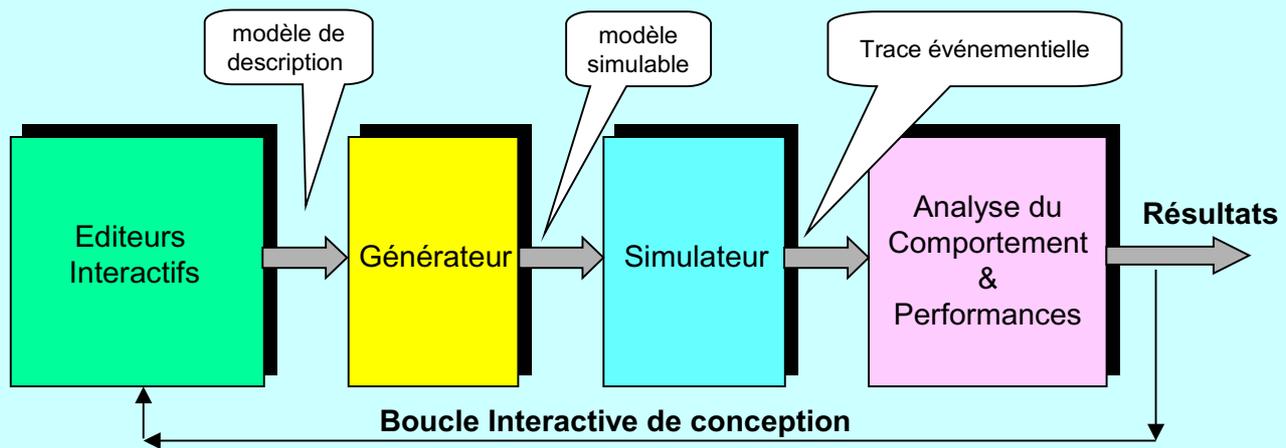
Function - Architecture CoDesign (VCC Cadence)



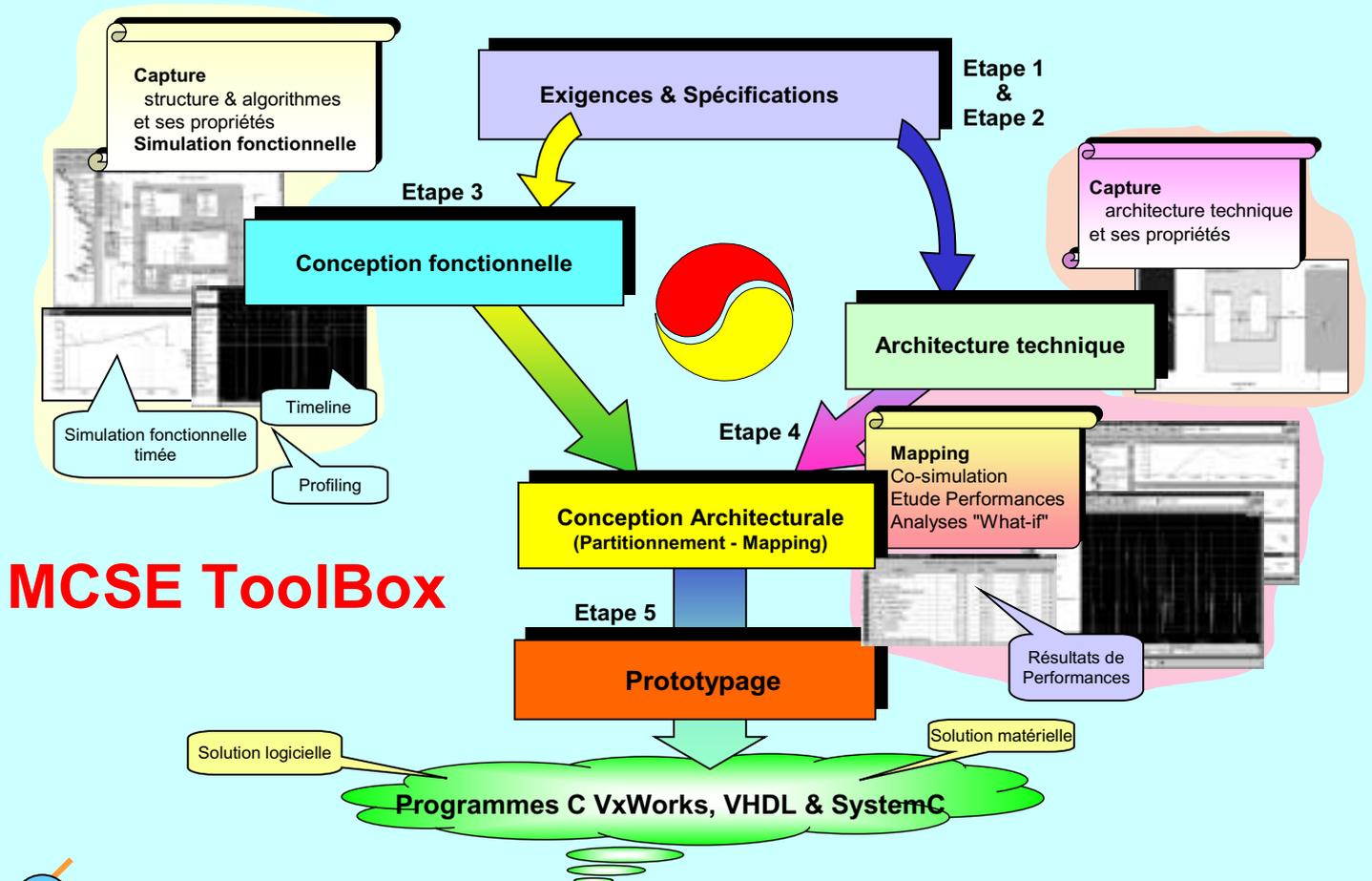
Notre approche : MCSE ToolBox



Cœur d'outils pour les 3 niveaux de conception



- Modèle C++ (et SystemC) simulable pour le niveau fonctionnel
- Modèle C++ (et SystemC) simulable pour le niveau architectural
- Prototype => Logiciel: C & RTK + matériel: VHDL synthétisable



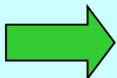
MCSE ToolBox

Caractéristiques de MCSE ToolBox

- Outil interactif simple d'utilisation
- Description séparée des 2 types d'architectures
- Partitionnement et allocation en interactif
- Simulation, observation, des propriétés fonctionnelles et architecturales
- Capture de différentes architectures pour l'exploration de l'espace de solutions
- Description à divers niveaux d'abstraction
- Evaluation des performances globales
- Génération de codes source C, SystemC, VHDL

Plan

- La conception système: c'est quoi
- Les bases de la modélisation et de la conception
- Spécification d'un système
- Conception fonctionnelle
- Conception architecturale
- Un outil comme support
- **Conclusions**

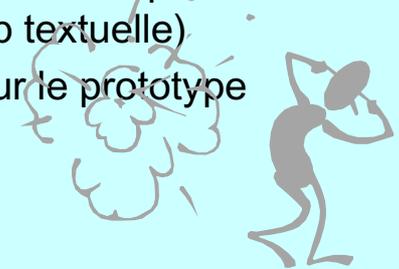


Conclusions (1/3)

- **Beaucoup d'évolutions attendues dans les 10 ans à venir**
 - technologies, outils, méthodes, organisations
- **Approche système inévitable**
 - approche par l'application
 - pas de séparation culturelle entre Hw et Sw
 - changement de technologies simple
 - réutilisation world-wide indispensable
- **Effort de formation important dans les entreprises**
 - progression ascendante lente, saut en abstraction
 - intérêt des jeunes déjà formés pour assurer la transition
- **Outils et IPs: évolution en cours**

Conclusions (2/3)

- **Problèmes de méthodologie :**
 - Séparation des compétences (Spécif. / Design / Hw /Sw)
 - Spécifications ne provenant pas (uniquement) des exigences client
 - Manque d'interaction ou mélange entre étapes
 - Manque d'utilisation d'outils durant la conception, surtout pour les premières étapes (spécification et conception trop textuelle)
 - Vérification des performances trop tardivement sur le prototype
 - Très peu d'exploration de l'espace des solutions
- **Problèmes des outils**
 - Outils commerciaux existants mais difficulté de les coupler dans un flot bien organisé
 - La réutilisation d'IPs pas suffisant: Capitalisation aussi et d'abord



Conclusions (3/3)

- Changement: du physique vers le fonctionnel

- Penser à des niveaux plus conceptuels



- Résultats escomptés d'une méthodologie système:

- Réduire l'effort d'apprentissage
- Augmenter l'efficacité en trouvant des solutions appropriées
- Réutiliser des connaissances et moins « ré-inventer la roue »
- Augmenter et canaliser la compétence de l'entreprise
- Augmenter le dialogue et la confiance des clients

- Difficultés

- « Not enough time » (résistance au changement), est un alibi
- Nécessité d'un « Coaching » pour favoriser le changement
- Investissement de long terme (on-the-job training)