

SA-RT
Structured Analysis for Real-Time

Jean-Philippe Babau

Département Informatique

INSA Lyon

SA-RT historique

Spécification de système

SADT

Structured Analysis Design Technics
(D.T. Ross 1976)

Spécification statique de logiciel

SA

Structured Analysis
(Yourdon/Demarco 1979)

Spécification dynamique de logiciel

SART

Structured Analysis for Real Time
(Ward/Mellor, 1985) (Hatley/Pirbhai, 1986)

« Stratégies de spécification des systèmes temps réel : SART »
D.J. Hatley, I.A. Pirbhai. Masson.

SA-RT objectifs

- **Document complet et lisible de descriptions des besoins**
- Outil de communication
 - formalisme graphique
 - client - concepteur
 - concepteurs
- Expression des besoins
 - fonctionnel
 - événementiel / réactif
 - données
 - architecture matérielle
- Vérification
 - cohérence
 - simulation

Principes de SA-RT

- Description de l'environnement : **diagramme de contexte**
 - **Bords**
 - Information échangées : **données** et **événements**
- Spécification des transformations de données : **analyse SA**
 - Aspect fonctionnel
 - flot de **données**
 - **processus** de transformation de données
 - Décomposition arborescente
 - Processus composites
 - Feuille : spécification des **processus primitifs**
- Description précise des données manipulées
 - **Dictionnaire de données**

Principes de SA-RT

- Description de l'aspect réactif : **analyse RT**
 - mise en évidence des **événements**
 - définition de la logique de contrôle : **processus de contrôle**
 - à un événement est associé un changement d'état et une ou plusieurs actions : **diagramme état-transition**
- Description de l'architecture matérielle
 - définition des machines et canaux de communication

Plan

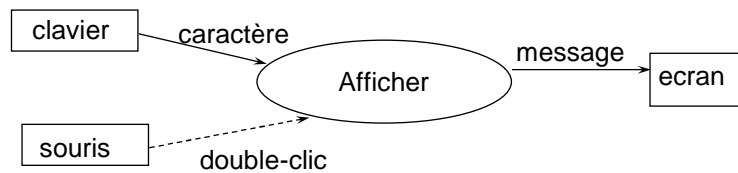
- Diagramme de contexte
- Analyse SA
- Analyse RT
- Méthodologie
- Architecture matérielle
- Exemple

Diagramme de contexte

- But
 - identifier les éléments qui communiquent avec le système
 - définir la liste des entrées/sorties du système
- Eléments représentés
 - bords
 - données et événements **utiles** échangés entre chaque bord et le système
 - **un** processus de transformation de donnée



Diagramme de contexte : exemple



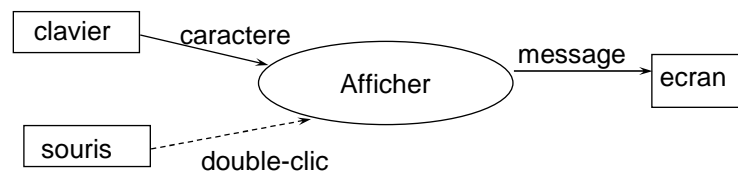
Type de bords

- Dispositif ou périphérique
 - Lecteur Code Barre, GPS, etc.
 - Capteur de position, capteur de température, etc.
 - Moteur électrique, vérin, etc.
- Élément de l'IHM
 - Clavier, écran, souris, joystick
 - Boutons
- Equipement physique
 - Bras de robot, convoyeur, ABS, TV, etc.
- Machine
 - serveur, superviseur, console opérateur, etc.

Description des bords

- Description précise de ou des entités modélisées
- Ne pas décrire dans un bord
 - Les données/événements fournies ou reçues
 - Le rôle de l'application
 - Éviter les verbes
- Ne pas confondre « bords » et « vue du bord au sein du système »
- Exemples
 - Un bouton est un élément de l'IHM
 - Il est caractérisé par sa couleur, sa forme, son type, son label, sa position
 - Un bouton n'est pas caractérisé par son utilité, son rôle
 - Un écran
 - Il est caractérisé par le nombre de lignes et de colonnes
 - Il n'est pas caractérisé par ce qui est affiché dessus

Diagramme de contexte : exemple



Description des bords

*souris : souris sans fil
clavier : clavier de type azerty
écran : écran plat 14 "*

Données et événements

- Flot de données
 - continu
 - discret
- Flot d'événement
 - continu
 - discret

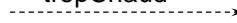
température



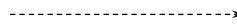
login



tropChaud



alarme



Description des données et des événements

- Donnée
 - Description informelle
 - Type : énuméré, réel, entier
 - Intervalle : [0,10]
 - Unité
- Événement
 - Description informelle
- Exemples
 - tensionMoteur : tension de commande du moteur, entier, [0,5], Volt
 - boutonXAppuyé : le bouton X a été appuyé

Dictionnaire de données

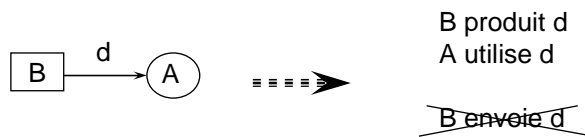
- Description des données et des événements
 - nom insuffisant
 - type et unité
 - intervalle ou valeur
 - opérateurs de composition
- Formalisme

| Symbole | Signification |
|---------------|----------------------|
| = | composé de |
| * * | commentaire |
| + | regroupement |
| { } | itération non bornée |
| n { } p | itération de n à p |
| (.....) | optionnel |
| " " | expression littérale |
| [... ...] | ou exclusif |
| ALIAS | équivalent à |

DONNEES = * description *
 * type : ..., valeur ou intervalle : ..., unité : ... *

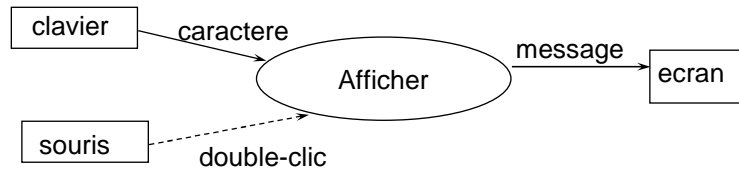
Description des données et événements

- Informations utiles
 - Pas d'aspects protocolaires



- Liens avec les bords
 - Données de « bas niveau »
 - Sorties : interprétables par le bord
 - Exemple pour un écran LCD
 - OK :
 - Erreur :
 - Entrées fournis par le bord
 - Exemple : pour un bouton
 - OK
 - Erreur :

Diagramme de contexte : exemple



Dictionnaire de données

caractere

code ASCII, char

message

chaîne de caractères terminée par un retour chariot, 1{char}255

double-clic

double clic gauche

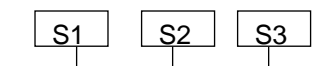
Regroupement des bords

- Regrouper les entités physiques ou logiques
- Logiques de regroupement
 - Entrées / Sorties
 - Par dispositif
 - Répartition géographique



Ci : capteur

Capteurs

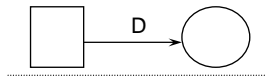


Si : serveur

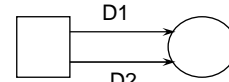
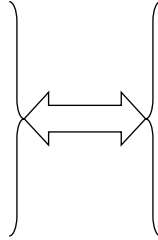
Serveurs

Regroupement des données et des événements

- Regroupement « et »

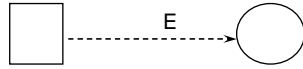


$D = D1 + D2$
D1 : donnée ...
D2 : donnée ...

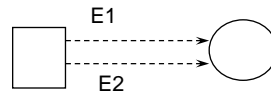
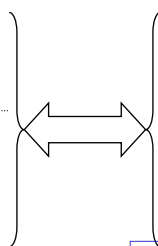


D1 : donnée ...
D2 : donnée ...

- Regroupement « ou »



$E = [E1 | E2]$
E1 : événement ...
E2 : événement ...



E1 : événement ...
E2 : événement ...

exemple

Cahier des charges

afficher la vitesse sur l'écran du tableau de bord lorsqu'on appuie sur le bouton « Speed »

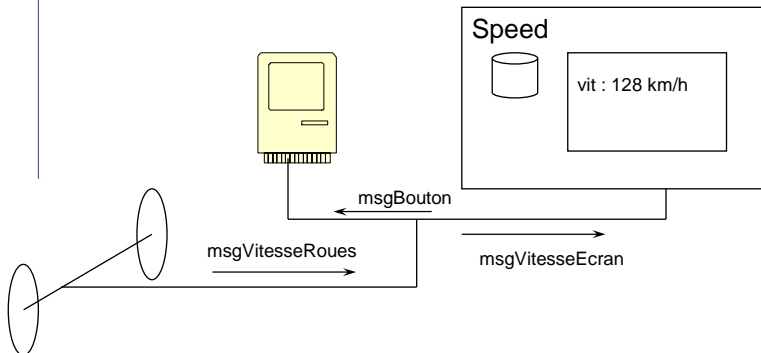
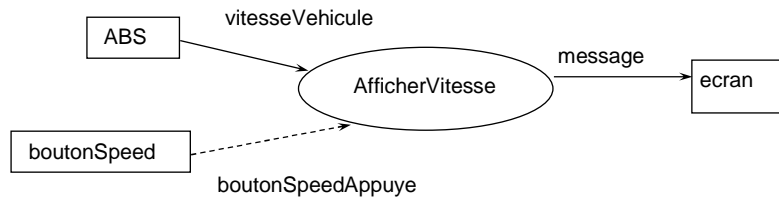


Diagramme de contexte : exemple



Description des bords

ABS

capteur de vitesse lié à l'ABS

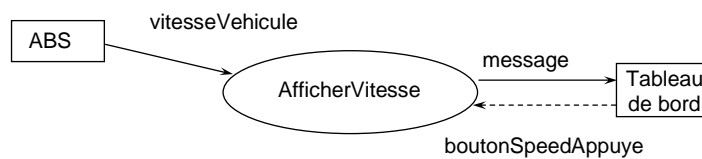
boutonSpeed

bouton situé en haut à droite sur le tableau de bord, label "Speed"

ecran

écran LCD 5 lignes (de 0 en haut à 4 en bas), 16 colonnes (de 0 à 15 de gauche à droite)

Diagramme de contexte : exemple



Description des bords

ABS

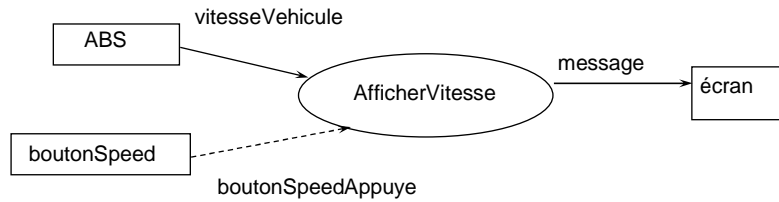
capteur de vitesse lié à l'ABS

Tableau de bord

boutonSpeed : bouton situé en haut à droite sur le tableau de bord, label "Speed"

écran : écran LCD 5 lignes (de 0 en haut à 4 en bas), 16 colonnes (de 0 à 15 de gauche à droite)

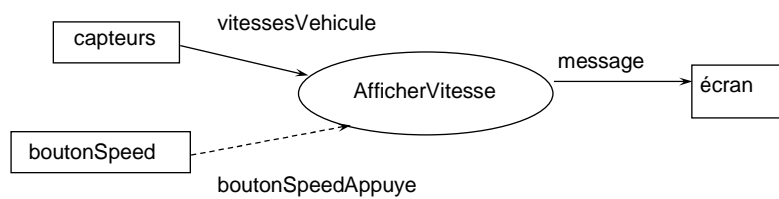
Diagramme de contexte : exemple



Dictionnaire de données

vitesseVehicule : vitesse du véhicule, type : réel, intervalle : [0.0,200.0], unité : km/h
boutonSpeedAppuyé : le bouton "Speed" est appuyé
message : chaîne de 16 caractères (n°colonne)

Diagramme de contexte



Description des bords

Capteurs

ABS de la roue droite et ABS de la roue gauche

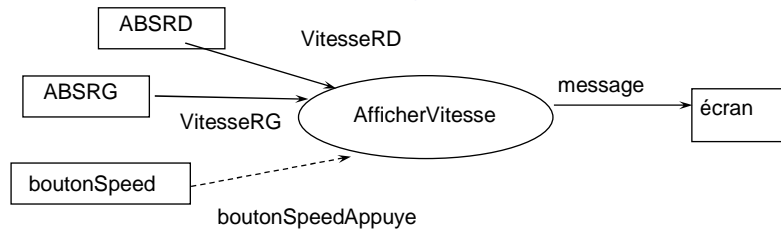
Dictionnaire de données

vitessesVehicule : VitesseRG + VitesseRD

VitesseRG : vitesse de la roue gauche, type : réel, intervalle : [0.0,200.0], unité : km/h

VitesseRD : vitesse de la roue droite, type : réel, intervalle : [0.0,200.0], unité : km/h

Diagramme de contexte



Description des bords

ABSRG

ABS de la roue gauche

ABSRD

ABS de la roue droite

Dictionnaire de données

VitesseRG : vitesse de la roue gauche, type : réel, intervalle : [0.0,200.0], unité : km/h

VitesseRD : vitesse de la roue droite, type : réel, intervalle : [0.0,200.0], unité : km/h

Diagramme de contexte

- Lisibilité
 - Regroupements
 - Nommage
- Descriptions
 - concise **et** précise
 - Un bord ≠ une donnée ≠ un événement ≠ une fonction
- Choix de représentation
 - Peuvent être différents
 - Homogénéité des choix

Plan

- Diagramme de contexte
- Analyse SA
- Analyse RT
- Méthodologie
- Architecture matérielle
- Exemple

Diagramme de Flot de Données (DFD)

- Définition
 - associé à un processus de transformation de donnée
 - le premier est associé au processus défini dans le diagramme de contexte
 - structuration hiérarchique

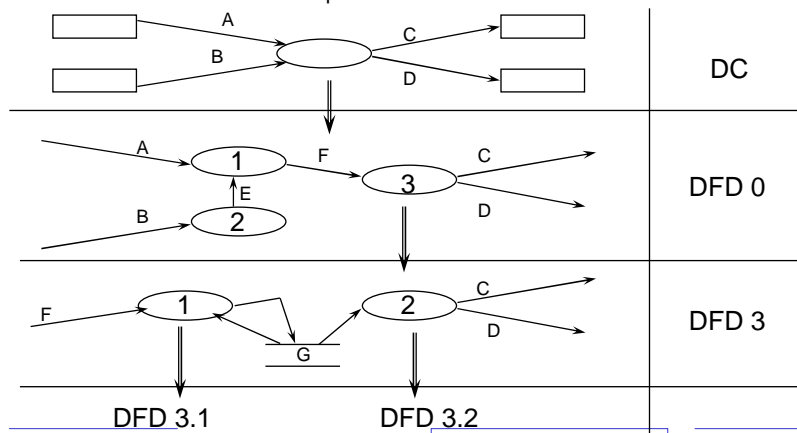
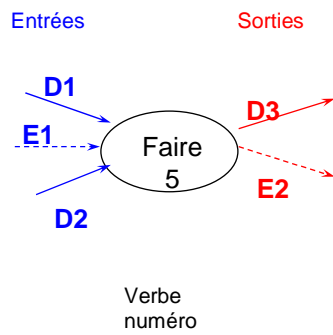


Diagramme de Flot de Données (DFD)

Processus de transformation de données

Vue « boîte noire »
« signature » de l'action Faire



DFD associé à Faire

Vue « boîte blanche »
« corps » de l'action Faire

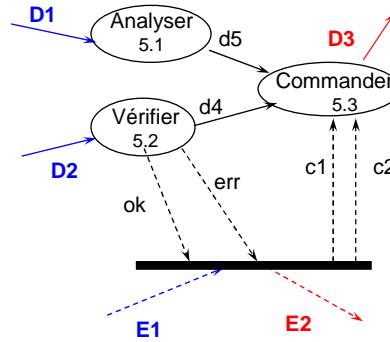


Diagramme de Flot de Données (DFD)

Transformation de données (SA)

- les flots entrants ou sortants
 - issus du processus de transformation de donnée

- des processus de transformation de données

- des flots de données d4 →

- des stocks de données — nom —

Verbe
numéro

Partie réactive (RT)

- un processus de contrôle —

- des flots d'événements (liés au processus de contrôle)

c1 →

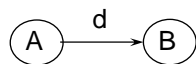
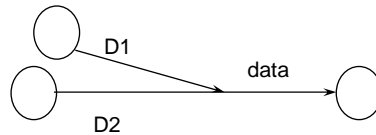
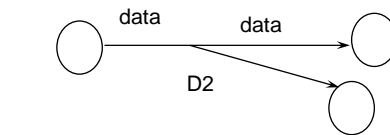
Les processus de transformation de données


- Représentation
 - symbole graphique
 - *Spécification informelle*
 - Non obligatoire
 - **Verbe**
 - Pour une action, une fonction
 - numéro
 - 0 : diagramme de contexte
 - 1..n : processus




Les données

- Dictionnaire de donnée
 - Cf. ci-avant
- Regroupement / Découpage
 - Cf. ci-avant
- Logique flot de donnée



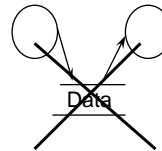
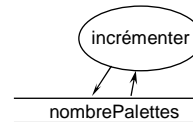
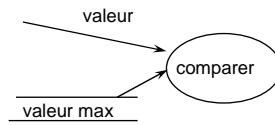
A produit d 

B utilise d 

~~A envoie d~~

Les stock de données

- Les constantes
 - Données numériques
 - Messages utilisateurs
- Les informations à modifier
 - Numéros, tables, compteurs, ...
- Un stock de donnée n'est pas
 - une variable de programmation
 - un flot de donnée
- Description : idem données

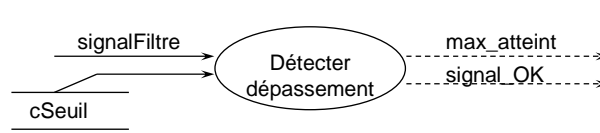


type de Processus

- Traiter des entrées
 - Conditionner
 - Filtrer
 - Analyser
 - Décoder
 - Interpréter
- Opérations
 - Calculer
 - Evaluer
- Traiter des sorties
 - Formater
 - Commander
 - Lisser
 - Afficher
- Gestion d'une fonctionnalité
 - Surveiller
 - Vérifier
 - Stocker
 - Compter
- Gestion d'un équipement
 - Gérer Robot, etc.
- Processus primitif

Processus primitif

- DFD non nécessaire
 - le processus est déjà spécifié
 - fonction connue
 - spécifié dans un autre diagramme
 - les sorties peuvent être évaluées en fonction des entrées
 - **Pas d'information supplémentaire nécessaire**
 - **Evaluation simple**
 - **Pas d'événement en entrée**
- Exemple



Processus primitif

- spécification procédurale

```
IN : signalFiltre, cSeuil
OUT : signal_OK ,max_atteint
si (signalFiltre > cSeuil)
    alors max_atteint émis
    sinon signal_OK émis
fin si
```

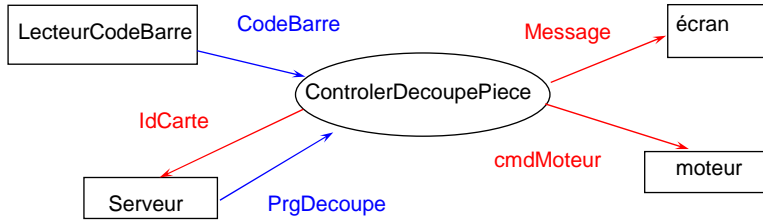
- spécification par précondition/postcondition

```
précondition : signalFiltre et cSeuil
postcondition :
    signalFiltre > cSeuil et max_atteint émis
    signalFiltre ≤ cSeuil et signal_OK émis
```

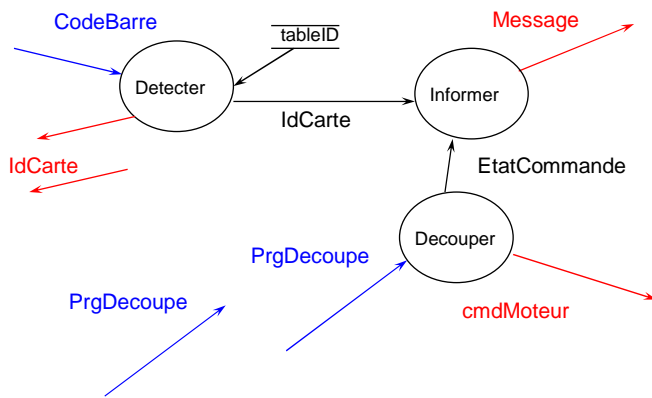
- spécification informelle (à éviter)

```
signal_OK émis lorsque signalFiltre est inférieur à cSeuil,
max_atteint sinon
```

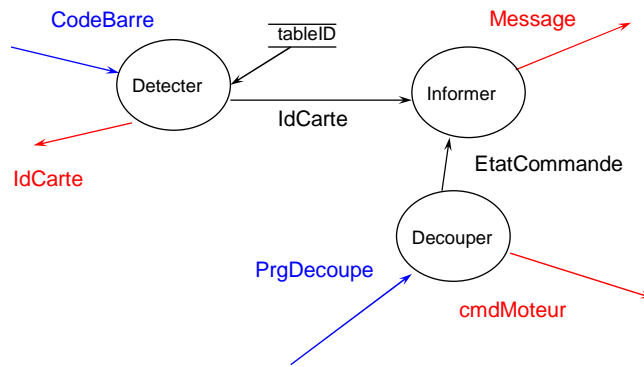
Exemple de décomposition



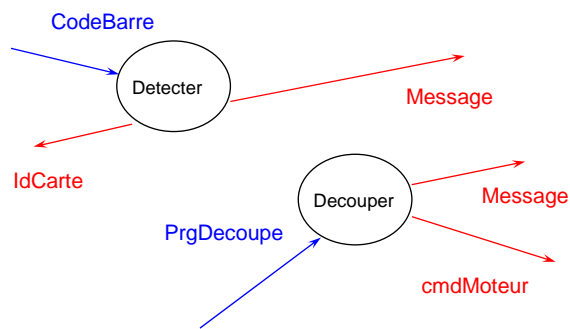
Exemple de décomposition



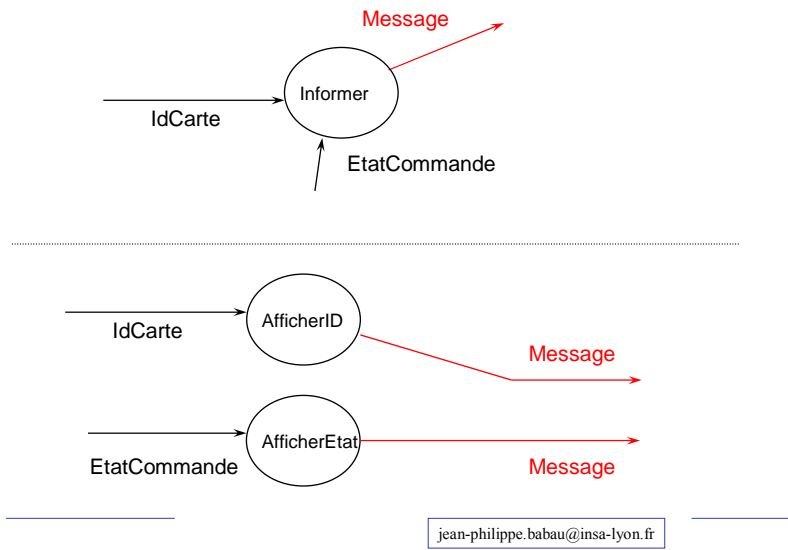
Exemple de décomposition



Exemple de décomposition

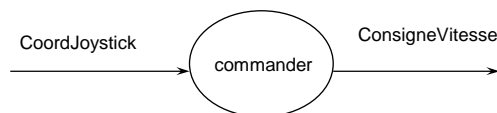
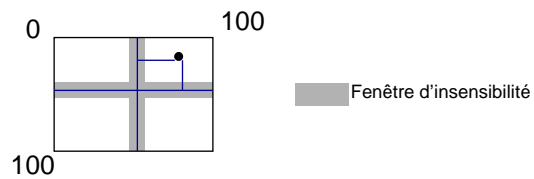


Exemple de décomposition



Exemple d'analyse SA

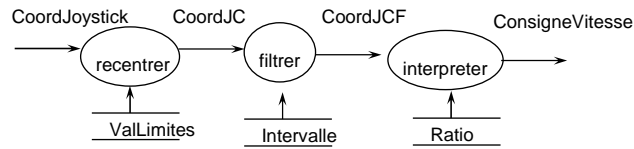
- Besoins
 - Production d'une consigne de déplacement pour un robot à partir d'une position de joystick



CoordJoystick : XJoystick+YJoystick
XJoystick : position en x, entier, [0,100]
YJoystick : position en y, entier, [0,100]

ConsigneVitesse : vitesseGauche + vitesseDroite
VitesseGauche : vitesse de la roue gauche, entier, [0,10]
VitesseDroite : vitesse de la roue droite, entier, [0,10]

Exemple d'analyse SA



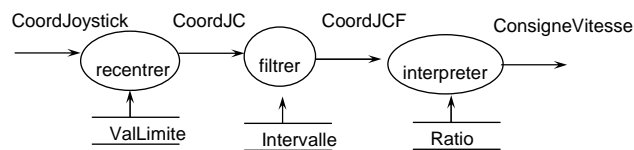
CoordJC : ALIAS CoordJoystick
CoordJCF : ALIAS CoordJoystick

ValLimites : $XE + YE$
XE : abscisse du point extrême en bas à gauche, 100
YE : ordonnée du point extrême en bas à gauche, 100

Intervalle : fenêtre d'insensibilité de 5

Ratio : coefficient de réduction de 5

Exemple d'analyse SA



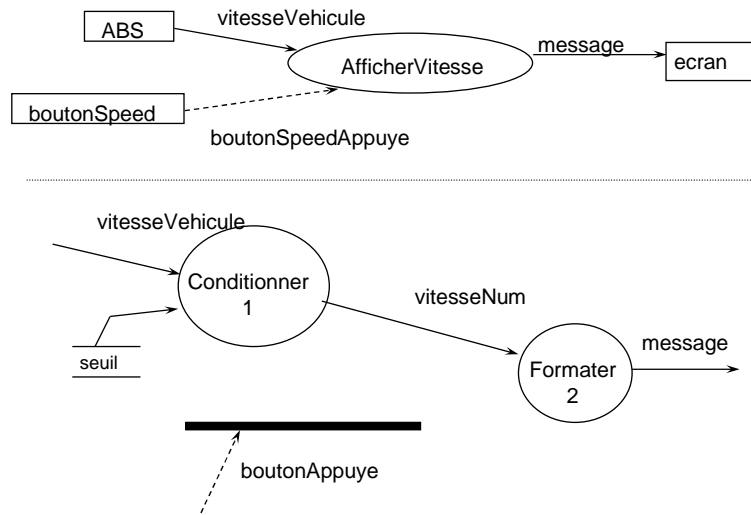
Processus primitifs

recenter
CoordJC. XJoystick = $\text{CoordJoystick.XJoystick} - XE / 2$
CoordJC. YJoystick = $YE / 2 - \text{CoordJoystick.YJoystick}$

filtrer
Si CoordJC. XJoystick < Intervalle
alors CoordJCF. XJoystick = 0
sinon CoordJCF. XJoystick = CoordJC. XJoystick
Idem pour CoordJC. YJoystick

Interpreter
 α : angle déterminé par le point = (CoordJCF. XJoystick , CoordJCF. YJoystick)
VitesseGauche = $\text{CoordJCF. XJoystick} \times (\cos(\alpha/2) / \text{Ratio})$
VitesseDroite = $\text{CoordJCF. YJoystick} \times (\sin(\alpha/2) / \text{Ratio})$

DFD 1er niveau



DFD 1er niveau

Processus primitifs

Conditionner

si $vitesseVehicule \geq seuil$ $VitesseNum = seuil$ sinon $vitesseNum = int(vitesseVehicule)$

Formater

message(ligne N^o1, à partir de la position 0) sur 14 caractères:

vit " " : " " VitesseNum sur 3 caractères " " km/h

" " : un espace; si VitesseNum sur moins de 3 caractères, on ajoute des espaces avant pour faire 3 caractères

Dictionnaire de données

Seuil : vitesse maximale affichée, réel, 150.0 km/h

VitesseNum : vitesse échantillonnée sur 2 octets, entier, [0, 150], km/h

Règles de représentation

- Egalité des flots entrants et sortants : processus associé / DFD
- DFD
 - processus : les actions et les fonctions du système
 - Des stocks de données : les constantes, les données du problème
 - **Un seul** processus de contrôle : la synchronisation des processus



Règles de décomposition

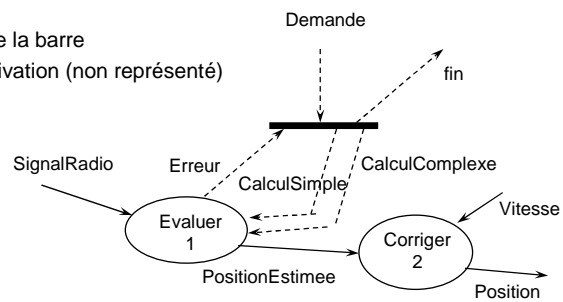
- Trouver les processus
 - Suivre des scénarios
 - Associer un processus à chaque donnée entrante et à chaque donnée sortante
 - Décomposition (filtrer / analyser) -> (Traiter) -> (restituer / formater)
 - Un processus par entité à contrôler
- Limiter le nombre de processus par niveaux
 - 5/6 au maximum
 - Regroupement logique
 - Plus de niveaux si nécessaire
- Masquer les stocks de données
 - Représenter uniquement les données partagées à un niveau
 - Constantes proches des processus primitifs
- Nommer correctement les processus
- Plusieurs stratégies possibles
 - Règles informelles

Plan

- Diagramme de contexte
- Analyse SA
- **Analyse RT**
- Méthodologie
- Architecture matérielle
- Exemple

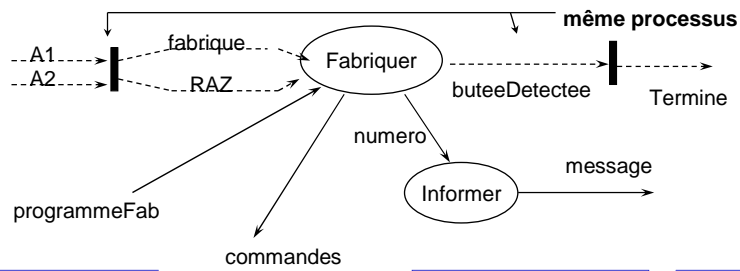
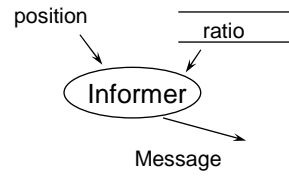
Processus de contrôle

- Découplage fonctionnel / événementiel
- Gestion centralisée par DFD de la synchronisation des processus
 - **Processus de contrôle**
- événement entrant sur la barre
 - signalisation d'un processus
 - événement de l'extérieur
- événements sortant de la barre
 - activation / désactivation (non représenté)
 - actions à faire
 - signalisation



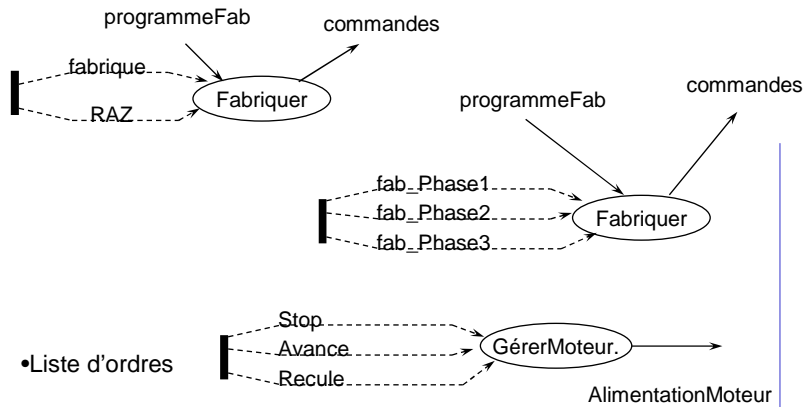
Les événements et le processus de contrôle

- Activation de processus primitif
 - activation non représentée
- un seul processus de contrôle



Les événements

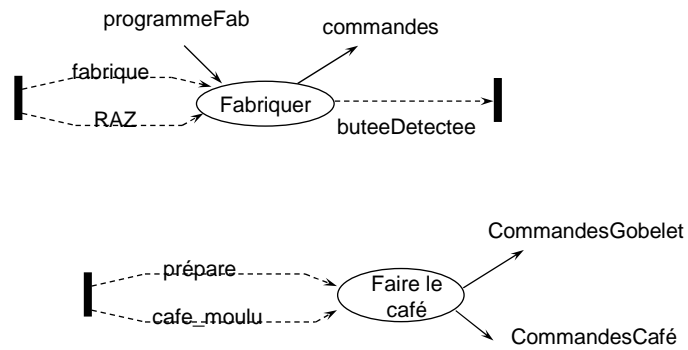
- Décomposition en sous actions



- Liste d'ordres

Les événements

- signalisation



Règles de représentation

- Egalité des flots entrants et sortants : DFD / processus associé
- **Un seul** processus de contrôle
 - synchronisation des processus
 - Tout événement passe par le processus de contrôle

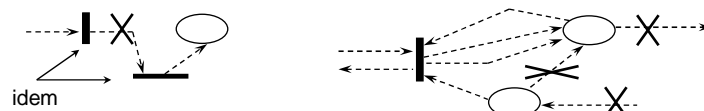
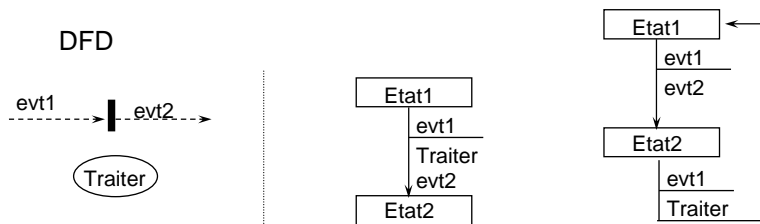


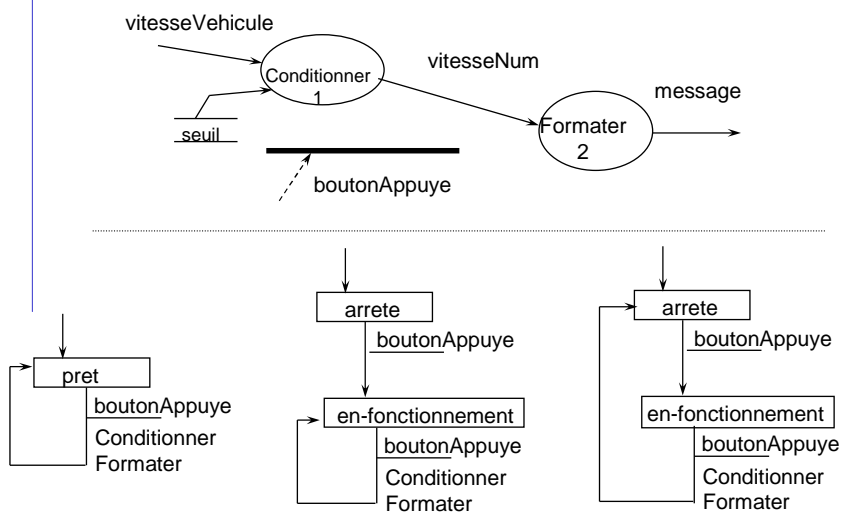
Diagramme état-transition : principes

- 1 DFD \Leftrightarrow 1 processus de contrôle \Leftrightarrow 1 diagramme état-transition
- Association
 - état - événement - action/signalisation
- Sens
 - Il est arrivé ceci : evt1
 - je suis dans un état : Etat1
 - je change d'état : Etat2
 - je peux activer un processus : Traiter
 - je peux signaler : evt2



jean-philippe.babau@insa-lyon.fr

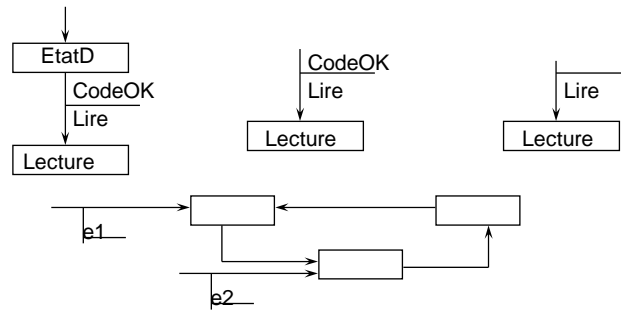
Exemple



jean-philippe.babau@insa-lyon.fr

Diagramme état-transition : états

- **A un instant donné**
 - le DFD est dans un état et un seul
- Un ou plusieurs états de départ



Démarrage

- Déclenchement
 - Branchement / démarrage du système
 - Reset
 - Séquence particulière
- Action d'initialisation
 - RAZ, configurations, ...

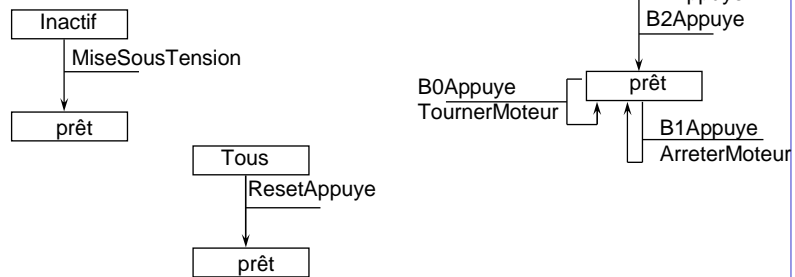


Diagramme état-transition : transitions

- Transitions
 - revenir dans le même état
 - plusieurs transitions sortent du même état
 - plusieurs transitions entre deux états

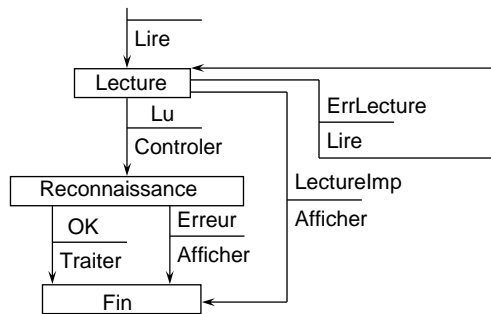
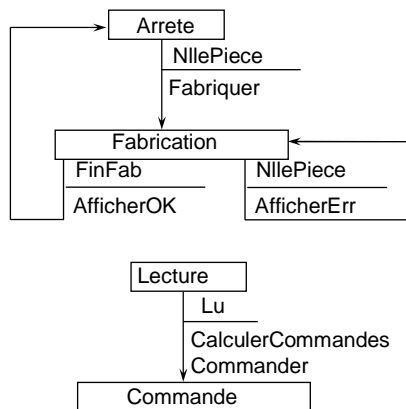


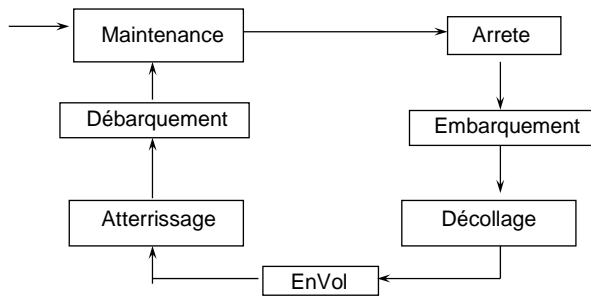
Diagramme état-transition : activations

- Activations
 - un même événement ne déclenche pas toujours la même action
 - plusieurs actions activées



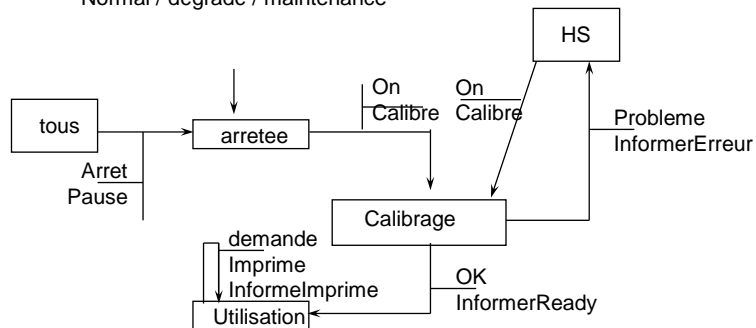
Cycle de vie de systèmes

- Phases
 - « Idle » / initialisation / fonctionnement / fin
- Mode de fonctionnement
 - Normal / dégradé / maintenance



Cycle de vie de périphérique

- Phases
 - « Idle » / initialisation / fonctionnement / fin
- Mode de fonctionnement
 - Normal / dégradé / maintenance



Stratégie de modélisation

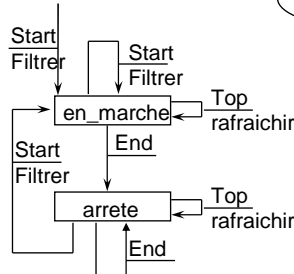
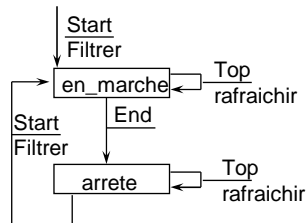
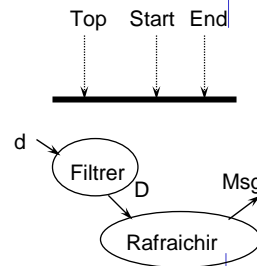
- Identifier les modes de fonctionnement
 - Marche, maintenance, ...
 - Relier les états
- Décrire le cycle de vie
 - Prêt -> initialisé -> enFonctionnement -> RAZ -> initialisé
- Suivre un scénario
 - A chaque nouvel événement : un « nouvel » état
- Se baser sur les états minimaux
 - Moins d'état mais perte d'information

Un état n'est utile au sens du comportement que si pour 2 événements distincts, il déclenche des événements distincts

- Plusieurs stratégies possibles
 - Nommage des états
 - Compromis exhaustivité / lisibilité

Principe de modélisation

- Dans chaque état
 - Décrire la réaction à tout événement possible
- Un événement non attendu
 - Est perdu : événement discret (start)
 - Est sauvegardé : événement continu (trop chaud)



Principe de modélisation

- Dans chaque état
 - Décrire la réaction à tout événement possible
- Un événement non attendu
 - Est perdu : événement discret (start)
 - Est sauvegardé : événement continu (trop chaud)

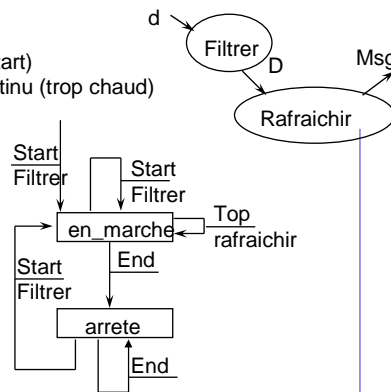
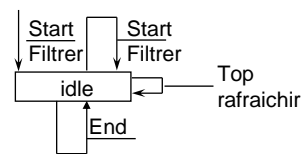
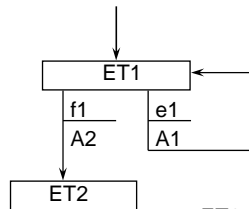
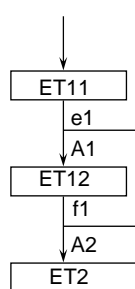


Diagramme état-transition : états

- Diagramme trop compliqué
 - reprendre le découpage des DFD
 - regroupement d'états



ET1 : ET11 ou ET12
e1 < f1

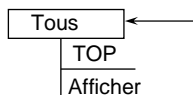
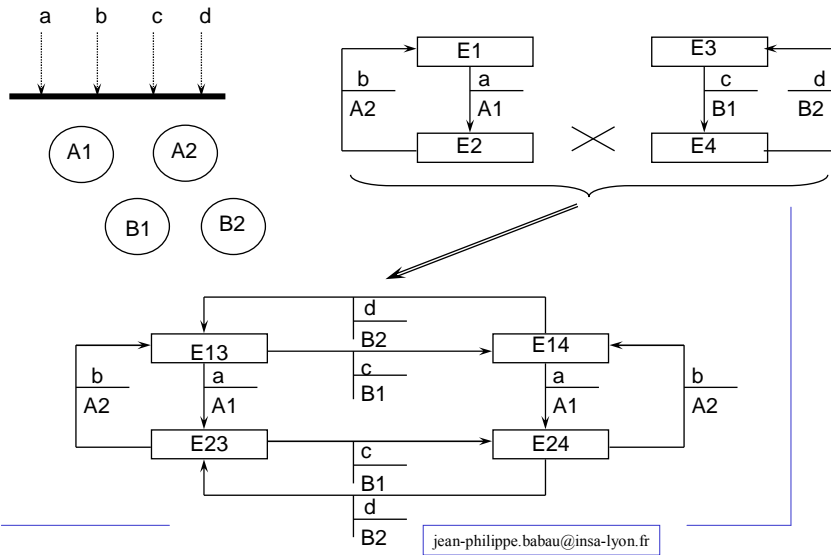
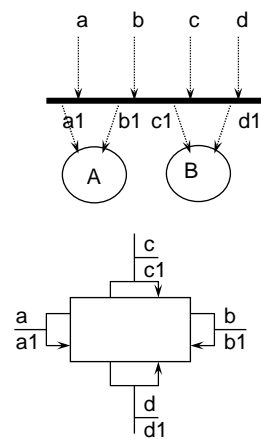


Diagramme état-transition : états

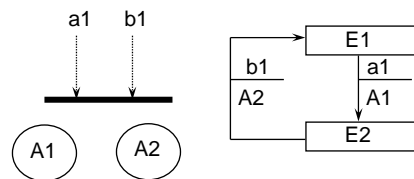


Réduction des automates

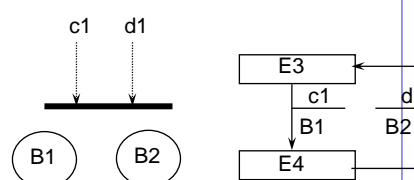
- Décomposer



DFD de A

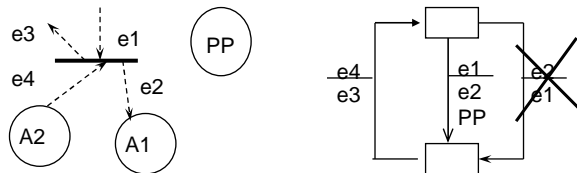


DFD de B



Règles de modélisation

- tout événement entrant dans le processus de contrôle
 - déclenche au moins une transition
 - ne peut pas être généré par le diagramme état-transition
- tout événement sortant de la barre de contrôle
 - est au moins généré une fois dans le diagramme état-transition
 - Ne peut pas déclencher de transition
- un événement entrant et sortant de la barre de contrôle ne peuvent pas avoir le même nom
- chaque processus primitif PP est activé au moins une fois dans le diagramme état-transition



Plan

- Diagramme de contexte
- Analyse SA
- Analyse RT
- **Méthodologie**
- Architecture matérielle
- Exemple

Méthodologie

- Diagramme de contexte
 - Liste des bords
 - Données et événements échangés
 - Spécification informelle (rôle)
- Pour chaque DFD
 - représenter les données entrantes et sortantes
 - Spécification informelle
 - Trouver les processus
 - Expliciter les données échangées
 - Décrire les stocks nécessaires
 - Constantes, données modifiées
- Dictionnaire de données
 - au fur et à mesure

Méthodologie

- Faire une analyse SA
 - découpage en sous actions
 - spécification informelle
 - données interne au DFD
- Faire une analyse RT
 - actions / sous-actions
 - besoins de synchronisation
 - diagramme état-transition en parallèle
 - Les états du système puis les transitions
 - Modélisation de scénarios puis complétude des événements
- Spécification des processus primitifs
 - Les sorties sont calculables en fonction des entrées

Méthodologie

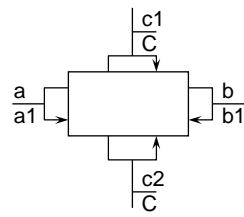
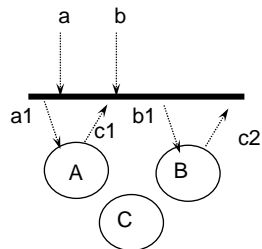
- Produire un document lisible
 - Choix des noms
 - Découpage en éléments simples
 - Processus primitif : **UN** algorithme
 - Décomposition guidée par la présentation
 - Plusieurs niveaux si nécessaire
- Produire un document complet et cohérent
 - Dictionnaire de données
 - Vérifier les flux entrants et sortants (DFD et processus de contrôle)
- Ne pas multiplier les descriptions
 - Choisir la **bonne** entité
 - Bord : élément qui communique avec le système de contrôle
 - Événement : « il s'est passé quelque chose »
 - Données : « information sur laquelle on peut faire des calculs »
 - Processus : la structuration / décomposition
 - Processus primitifs : les opérations sur les données
 - Réaction aux événements : diagramme état/transition

Quelques éléments pour la modélisation

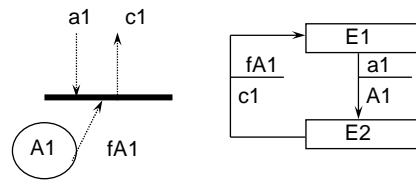
- Diagramme de contexte
 - Les sorties : préférer les données aux événements
- Le nommage : suivre un guide style
 - Toutes les constantes commencent par c
 - ...
- Analyse descendante
 - Un processus doit être self-consistent
 - Inutile de « remonter » pour comprendre
 - Éviter les événements en sortie de processus
 - Uniquement des signalisations

Hierarchiser le niveau d'abstraction

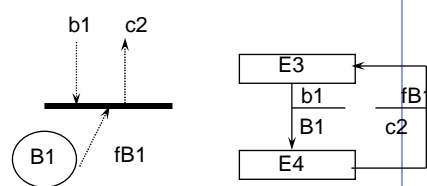
- DFD



DFD de A

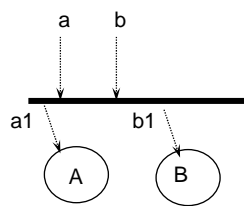


DFD de B

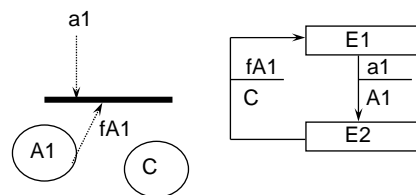


Réutilisation

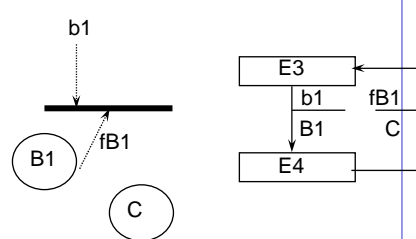
- DFD



DFD de A



DFD de B

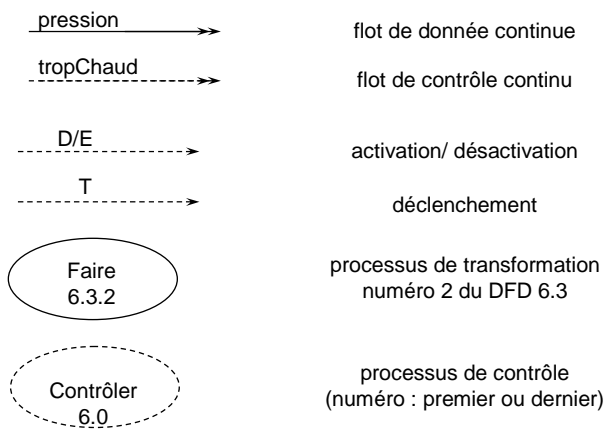


Conception

- Objectifs
 - Exécution des actions selon les besoins exprimés
- Activités du concepteur
 - Choix ou adaptation à une architecture matérielle
 - Mise en place des pilotes
 - Echanges de données et protocoles de communication
 - SART : spécification de la couche application
 - Choix d'un OS et définition du modèle d'exécution
 - Multitâches ou cyclique
- Paradigmes
 - SART
 - un événement => une séquence d'actions
 - Multitâches
 - Tâches (priorités), protection et partage des données
 - Protocole d'exécution des actions
 - Un processus ≠ une tâche, un processus primitif ≈ une fonction
 - Un événement ≠ un sémaphore booléen, une donnée SART ≠ mbx ou variable globale
- Méthode CODARTS pour le multitâches
 - Principes informel
 - un événement extérieur : une tâche
 - événements de même fréquence : une même tâche

Ward / Mellor

- Hatley / Pirbhai
- Ward / Mellor



Plan

- Diagramme de contexte
- Analyse SA
- Analyse RT
- Méthodologie
- **Architecture matérielle**
- Exemple

Architecture matérielle

- Définition de l'architecture matérielle
 - machines (unité de traitement)
 - interfaces matérielles
 - canaux de communication
- Placement
 - processus
 - données et événements
- Dimensionnement des machines
- Dimensionnement des réseaux
- Spécification des logiciels de communication

Diagramme de contexte d'architecture

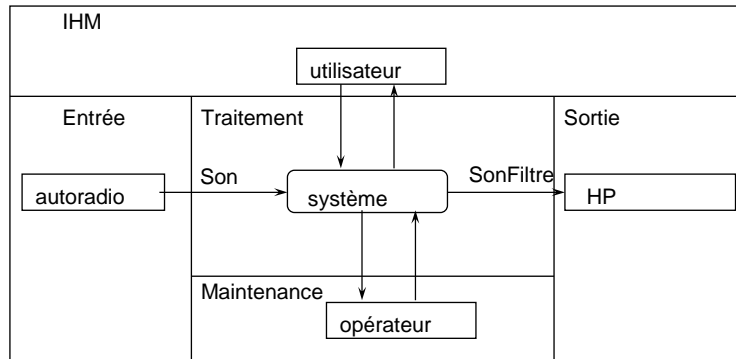
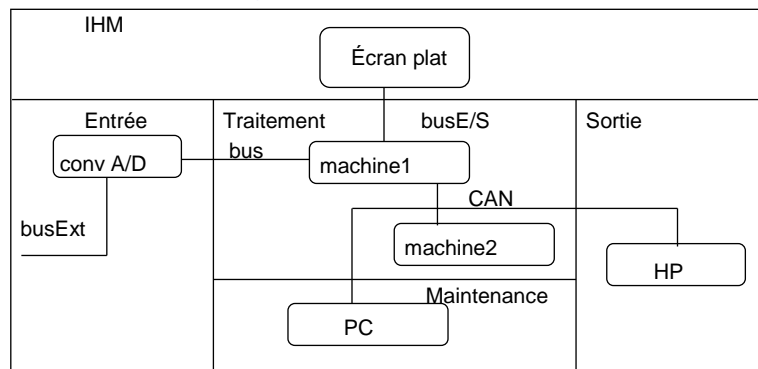


Diagramme d'interconnexion



- Types de lien de communication
 - bus
 - réseau
 - lien optique

Placement

- Processus de transformation de donnée (DFD)
 - module (machine)
- Processus de contrôle
 - module (machine)
- Données et événements entre deux modules
 - canal de communication
- Dimensionnement
 - machines
 - fonctionnalités + données
- Besoins en communication
 - pilotes
 - quantité de données échangées
 - protocoles

Contraintes temporelles

- Diagramme de contexte
 - Événements entrants et sortants
- Contraintes d'activation
 - Événement en entrée périodique (période)
 - Événement en entrée sporadique (dmin)
- Contraintes de régularité
 - Événement en sortie périodique (période)
 - Événement en sortie sporadique (dmin)
- Contraintes de délai
 - délai maximal de prise en compte d'un changement
 - Entre une entrée et une sortie

Conclusion

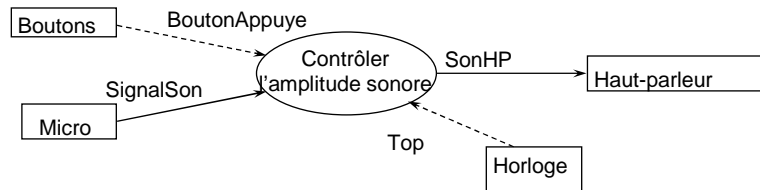
- Spécification des besoins
 - Fonctionnels
 - Réactifs
 - Matériels
 - Temps réel
- Exprimer des choix
 - Un document incomplet et difficile à lire : document **inutilisable**
 - Préciser la spécification
 - Complétude, correction, cohérence
 - Stratégie de représentation
 - Clarté, lisibilité

Plan

- Diagramme de contexte
- Analyse SA
- Analyse RT
- Méthodologie
- Architecture matérielle
- Exemple

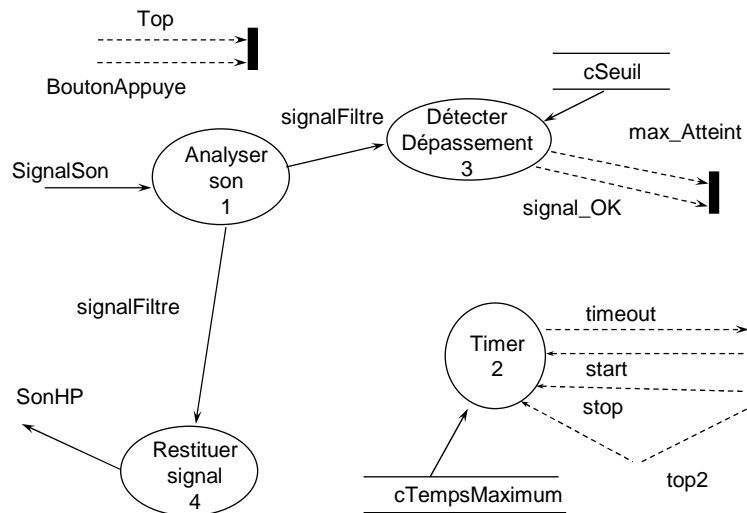
Exemple : diagramme de contexte

- Cahier des charges
 - le logiciel scrute le son en provenance d'un micro afin de l'envoyer vers un haut-parleur. Il coupe les sons trop forts. Le logiciel possède une commande marche/arrêt.
- Diagramme de contexte

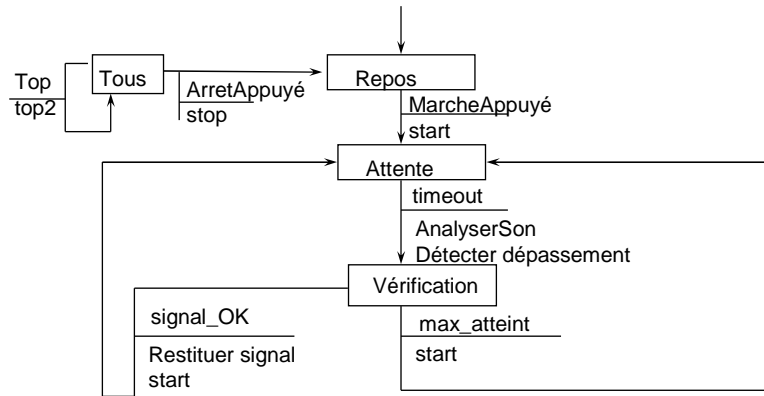


Boutons : bouton de label « Marche » et bouton de label « arrêt »
 Micro : microphone de marque XX
 Horloge : base de temps du système
 Haut-parleur : Haut Parleur situé sur le tableau de bord

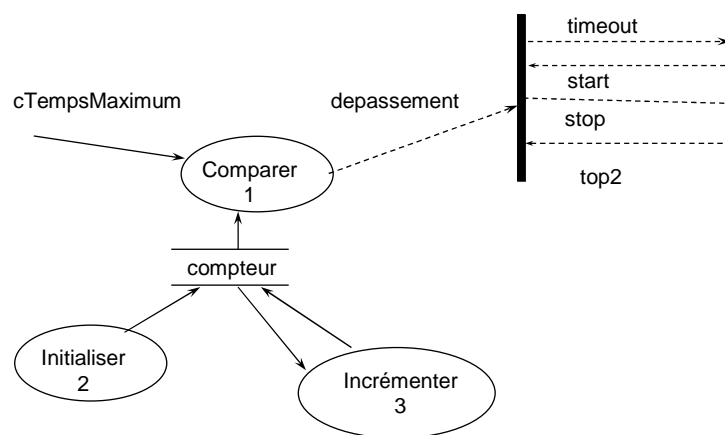
Exemple : DFD0



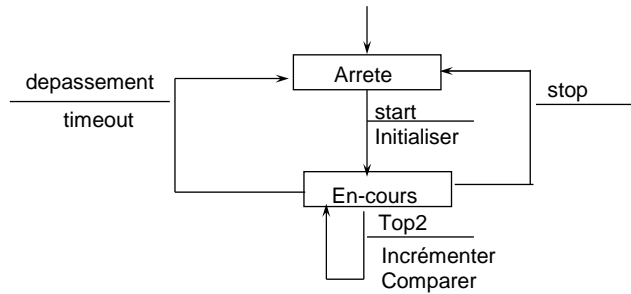
Exemple : diagramme état-transition



Exemple : DFD0



Exemple : diagramme état-transition



Exemple : spec des processus primitifs

Initialiser : *compteur* = 0

Incrémenter : *compteur* = *compteur* + 1

Comparer : si *compteur* > *cTempsMaximum* signaler *depassement*

Dictionnaire de données

| Nom | Description |
|--------------|--|
| SignalSon | * Signal analogique du son enregistré* |
| signalFiltre | * type : réel, intervalle : 0-300, unité : mVolt * |
| cSeuil | * son échantillonné * |
| SonHP | 8 { bit } 8 |
| max_atteint | * valeur maximale fixée pour le son * |
| signal_OK | * type : entier, valeur : 250 mVolt * |
| BoutonAppuye | ALIAS signalFiltre |
| MarcheAppuye | * niveau de son trop élevé * |
| ArretAppuye | * niveau de son correct * |
| | [MarcheAppuye ArretAppuye] |
| | * Appui sur le bouton « Marche » * |
| | * Appui sur le bouton « Arret » * |

Dictionnaire de données

| Nom | Description |
|---------------|--|
| Top | * Tick d'horloge * * Fréquence : 1 ms * |
| top2 | ALIAS Top |
| start | * Active le timer à 0* |
| stop | * Arrêt du timer * |
| timeout | * temps dépassé * |
| cTempsMaximum | * Temps d'attente maximal : 10 ms * |
| depassement | * Valeur supérieure au seuil * |
| compteur | * Nombre de top * * entier, [0,10] * |