

Modèles et Méthodes d'Analyse et de Conception

Types de modèles
Langage graphique

Le contenu est basé sur les transparents de la 10^{ème} édition
de “*Software Engineering*” de Ian Sommerville

L'Analyse

Modèles de système

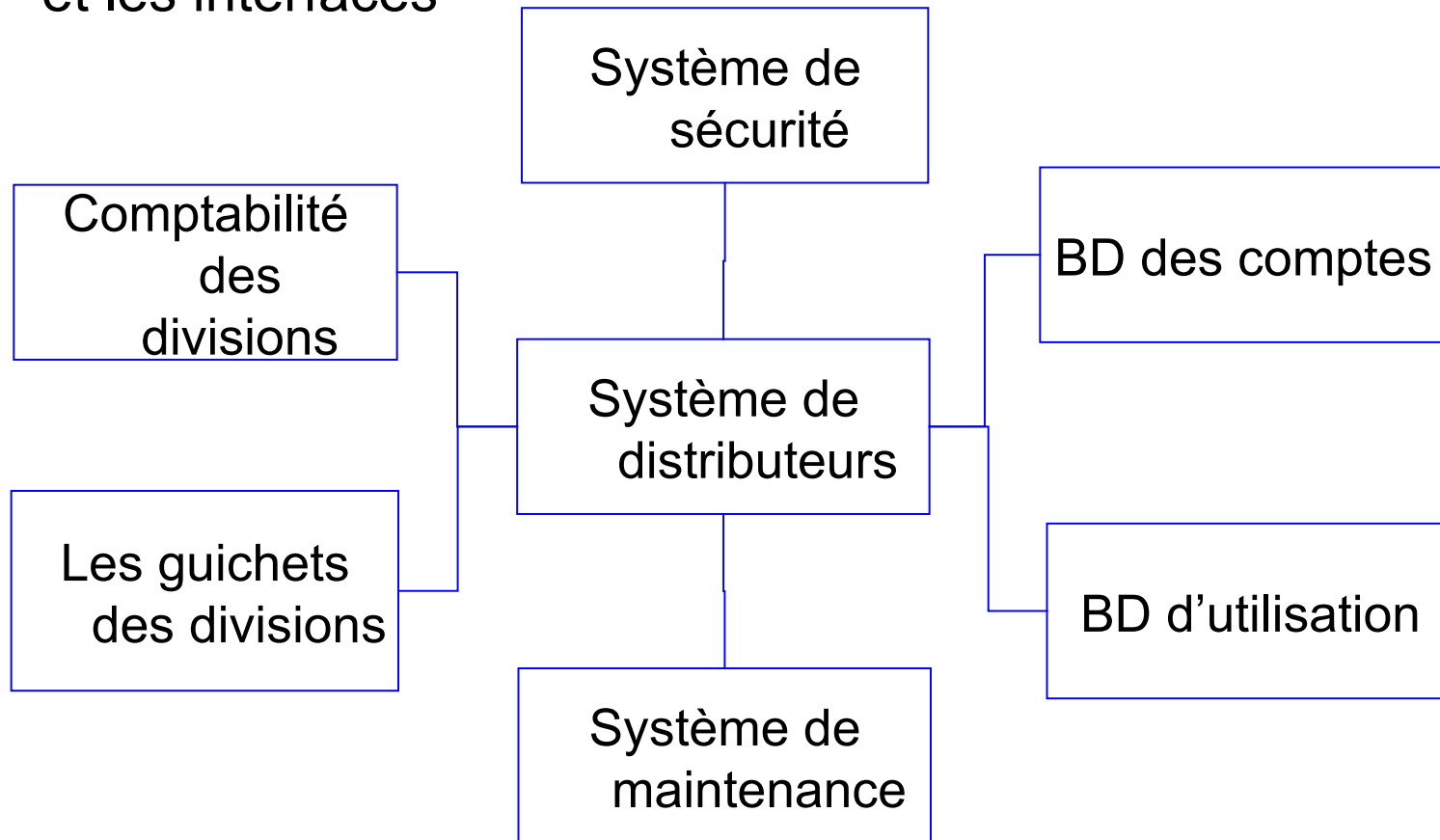
- Objectifs
 - Comprendre la fonctionnalité
 - Communiquer avec les clients
 - Documentation du logiciel
- Les perspectives
 - Externe
 - De comportement
 - Structurale
- Si on prend en compte le temps
 - Statiques
 - Dynamiques

Types de modèles

- Modèle de traitement des données
- Modèle de composition
- Modèle d'architecture
- Modèle des classes
- Modèle de stimulus et réactions

Modèle de contexte

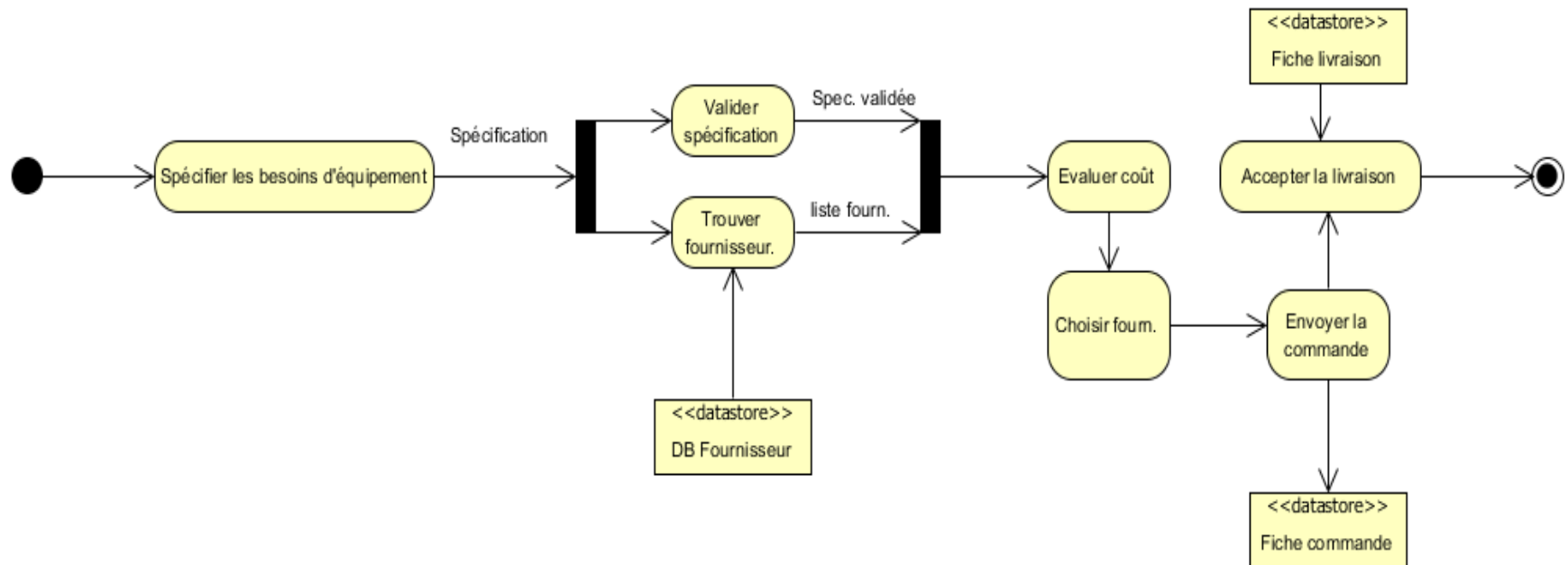
- Système de distributeurs de billets – ils spécifient les bornes et les interfaces



Modèle des processus

- Le processus générale et les sous-processus
- Il peut être fait par le DFD diagramme (flux de données)

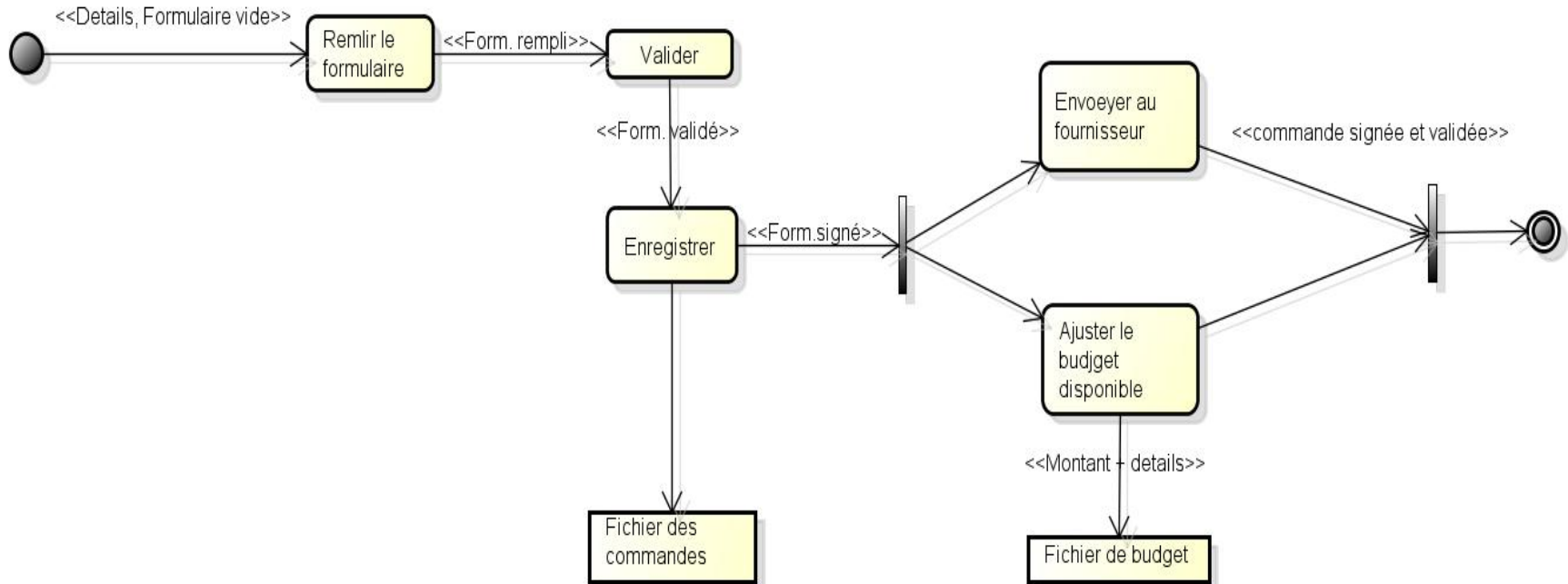
Commande et livraison d'équipement



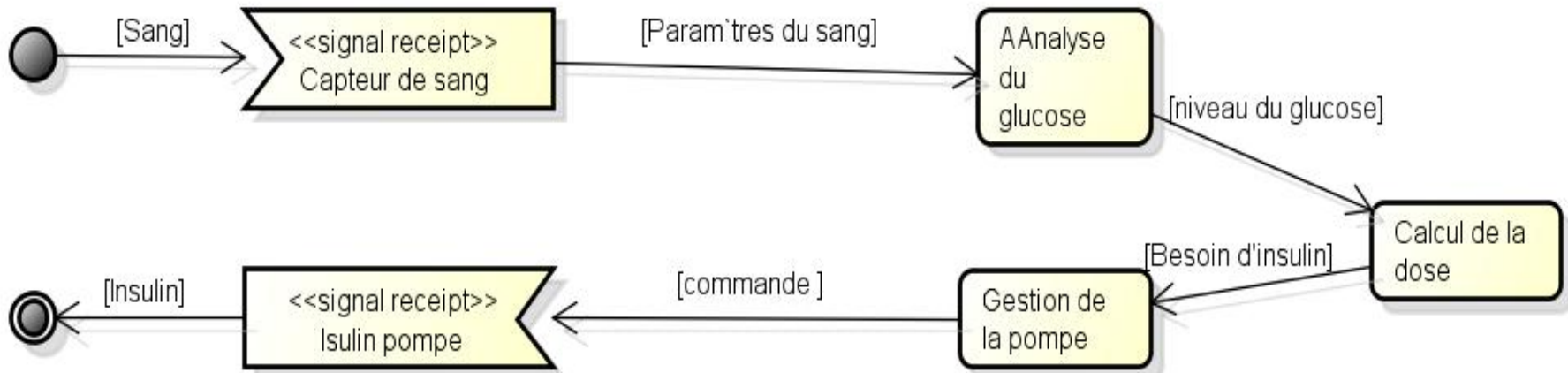
Modèles de comportement

- Types – deux perspectives différentes
 - Modèle de traitement des données
 - Modèle d'états
- Modèle de traitement des données
 - Diagramme des flux des données
 - Le traitement des données de début à la fin
 - On peut l'utiliser pour montrer l'échange de données avec les systèmes de contexte
 - Diagramme des activités (en UML)

Traitement d'une commande



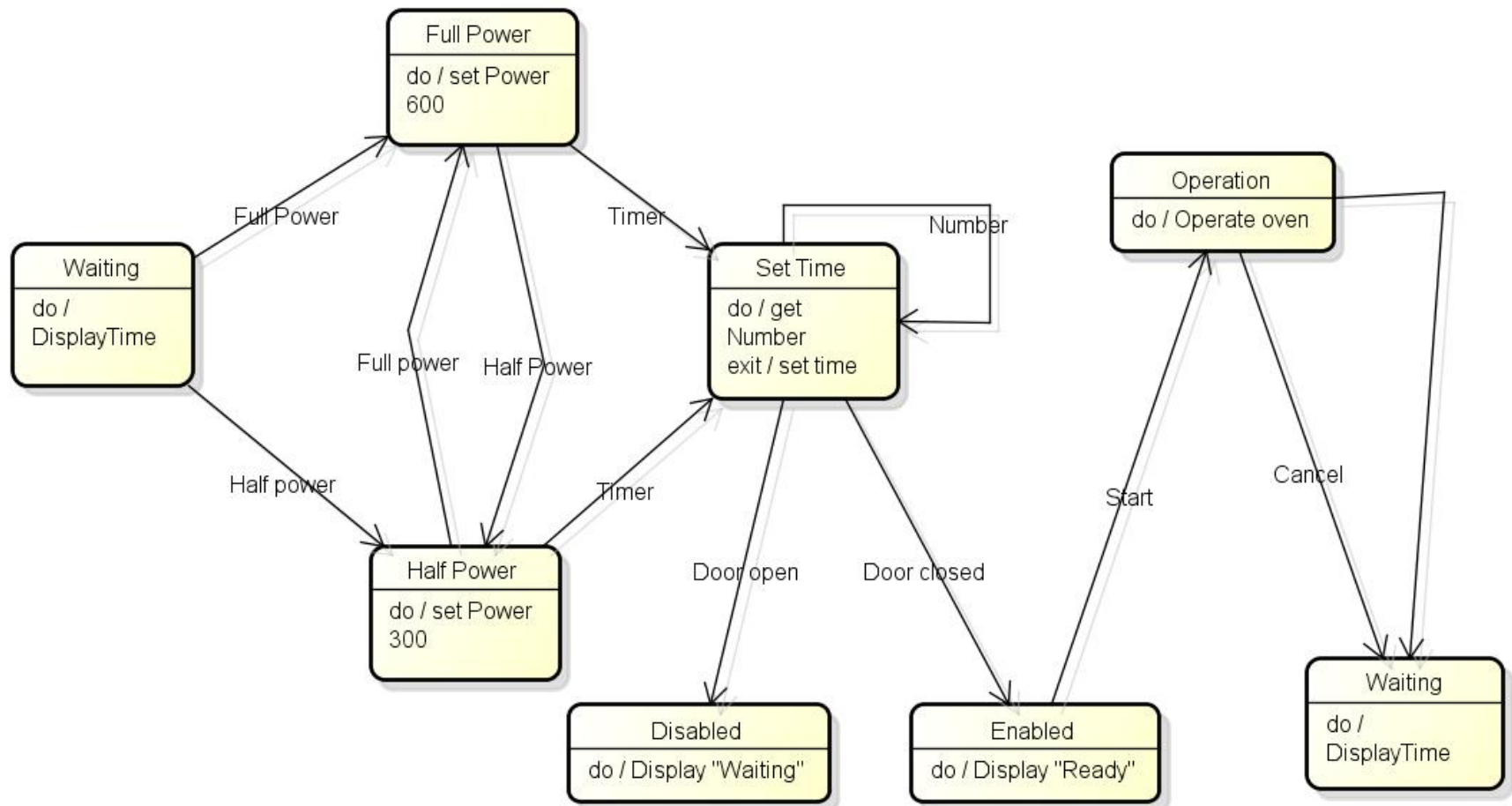
Le DFD de la pompe d'insulin



Le modèle d'états

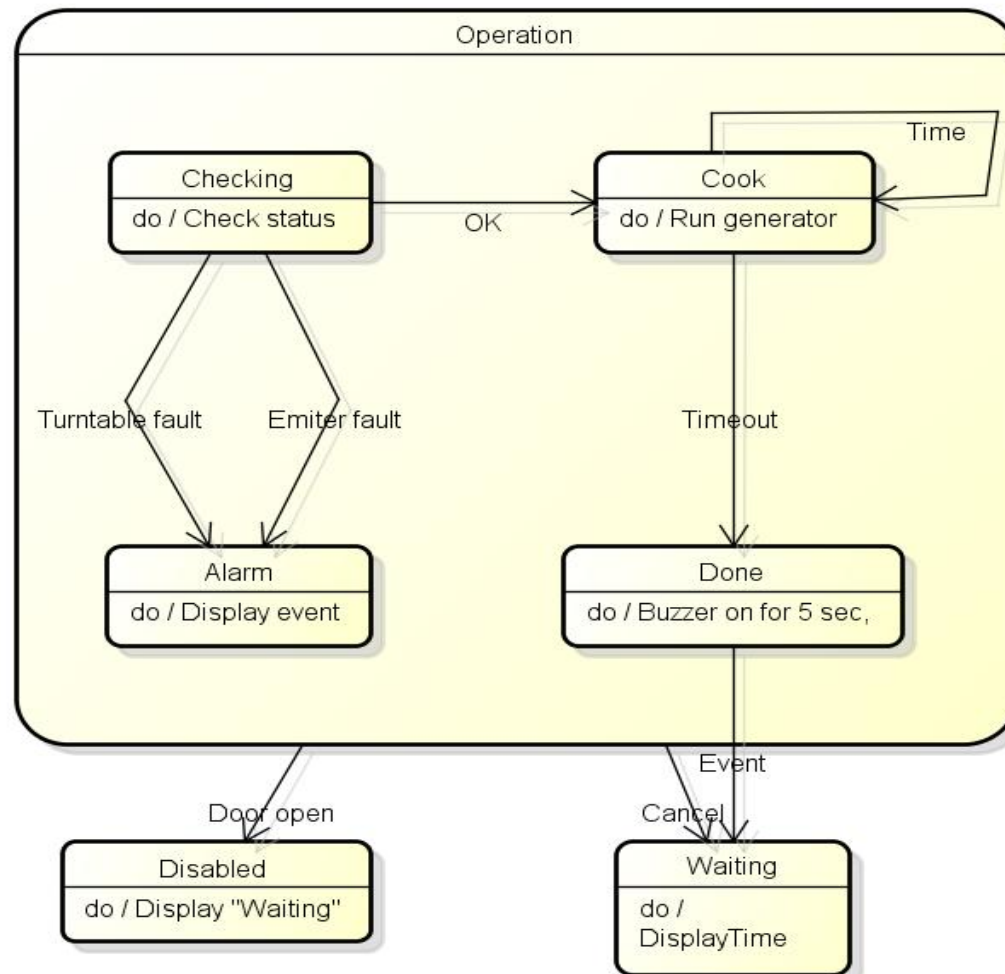
- Modèle qui démontre les réaction du système aux événements externes et internes.
- Il est appelé encore un automate fini.
- Il permet la décomposition aux sous-modèles

Le modèle d'un four micro-ondes



Le modèle d'un four micro-ondes

- Opération



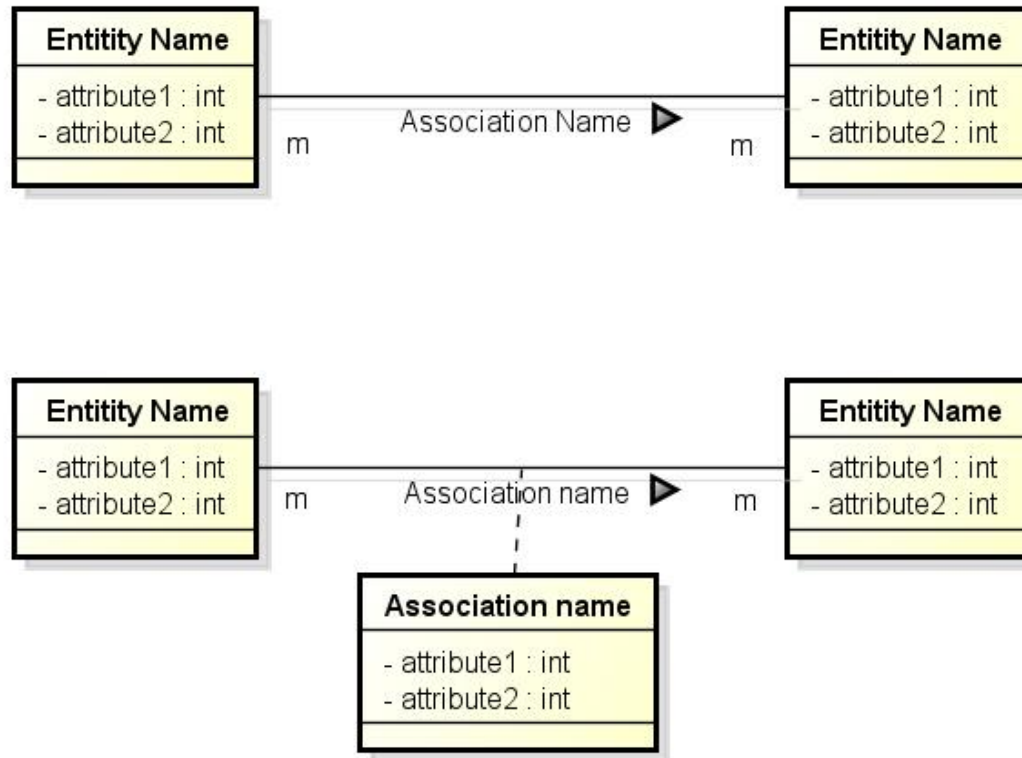
Le modèle d'un four micro-ondes

- Présentation comme tableau

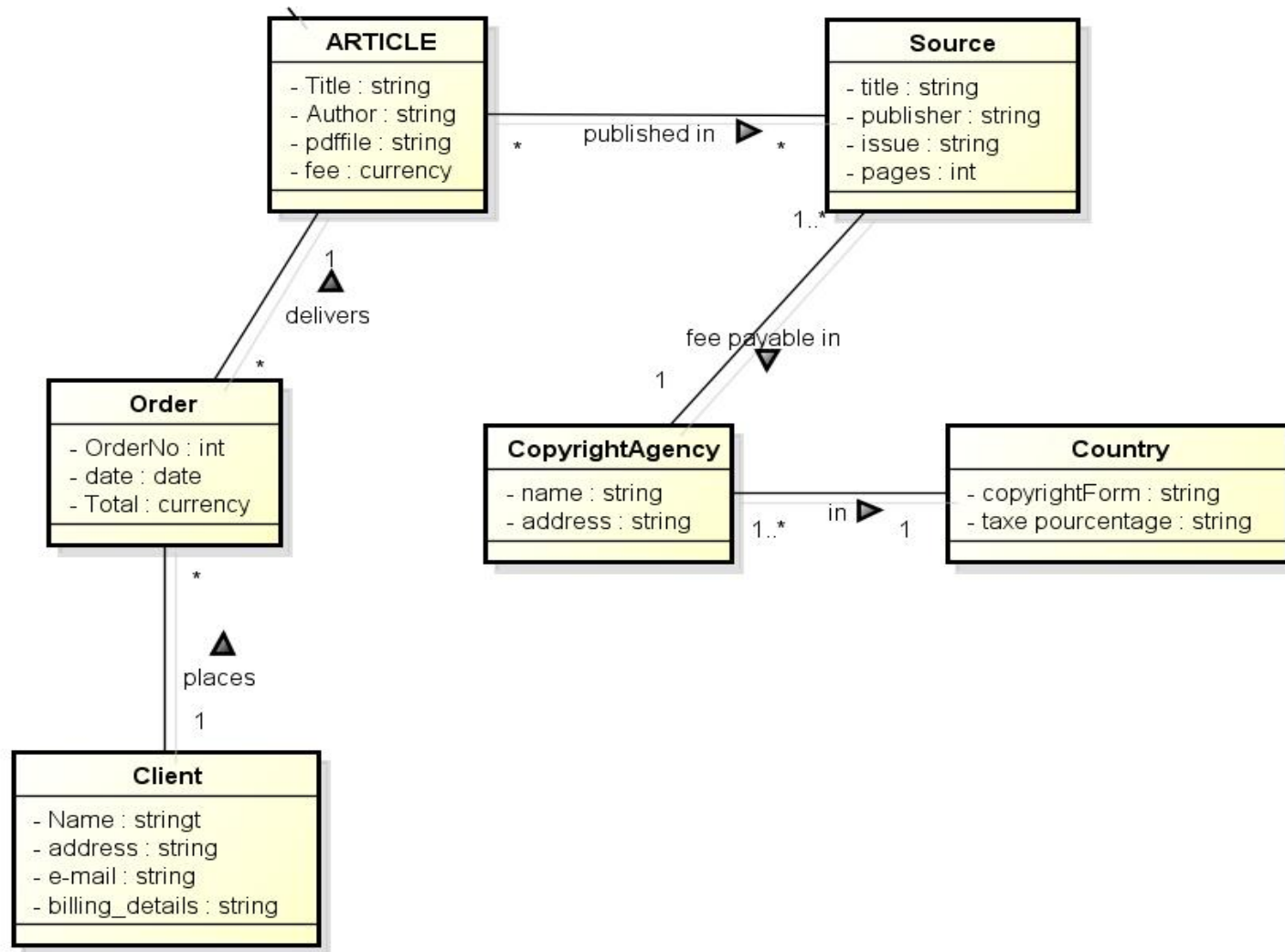
Etat	Description
Waiting	Le four attend la saisie. L'écran montre le temps
Half power	La puissance est fixée à 300 Watt. L'écran montre « Half power »
Full power	La puissance est fixée à 600 Watt. L'écran montre « Full power »
Set time	Le temps pour l'opération est entré par l'utilisateur. Le temps est montré à l'écran.
.....	
.....	

Modèle sémantique des données

- En fait c'est le modèle conceptuel des données
- On utilise le diagramme des classes



Modèle des données de LYBSYS



Dictionnaire des données

- C'est une liste des données avec leurs noms et description (voir les BD)
- On peut inclure et les descriptions des entités, associations et attributs.
- Objectifs
 - Ensembles des noms du système
 - Eviter les duplicats
 - Aider l'analyse

Les modèles objets

- Ils présentent le système ou différentes parties comme ensembles des objets avec leurs membres leurs fonctionnalités et les associations entre eux.

Les modèles objets (UML)

- Types d'associations
 - Héritaire
 - D'agrégation
 - De dépendance
 - D'interaction (utilisation)
- Notation des contraintes
 - {spécification de la contrainte}
 - Types
 - Langue naturelle
 - Pseudo code
 - Expression mathématique

Modélisation du comportement des objets

- Diagramme des séquences
- Diagramme d'activités
- Diagramme des états
- Diagramme de communication

Les faiblesses des modèles

- Les besoins non-fonctionnels
- Trop détaillés et incompréhensibles.

Les Méthodes d'analyse et de conception

Dans les méthodes on retrouve les concepts de base

- la construction de modèles**
- la description du général au particulier**
- la recherche d'une solution d'ensemble**
(données, traitements, organisation de travail, etc.)
- la préoccupation constante pour une solution de qualité**
- l'adaptabilité de la solution implantée par rapport à l'évolution de son environnement**
(mission, lois et règlements, nouvelles technologies)

Les grands types de méthodes d'analyse et de conception

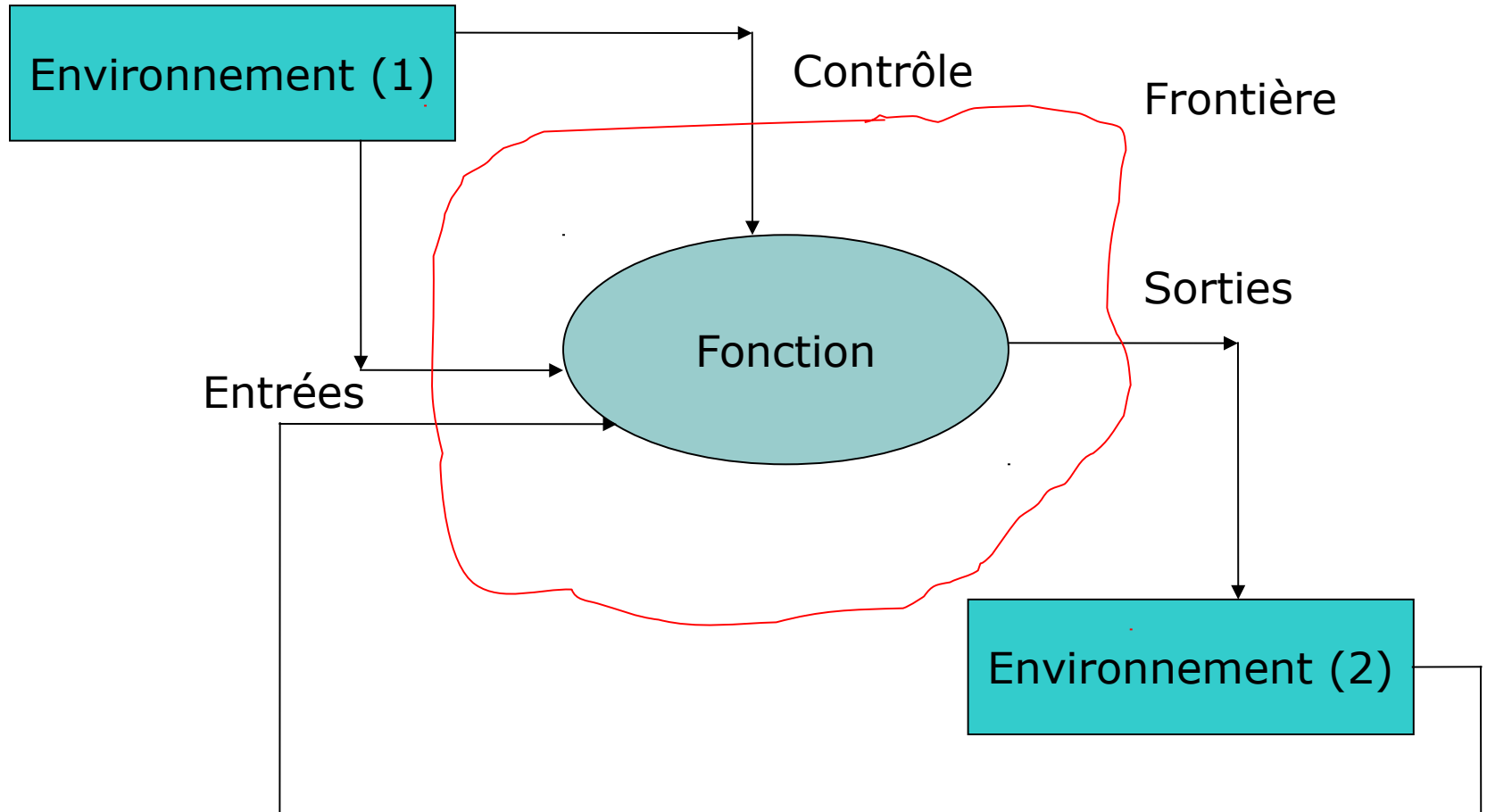
- ❑ **Méthodes Fonctionnelles** (SADT, DFD)
- ❑ **Méthodes Systémiques** (Merise)
- ❑ **Méthodes Dynamiques** (SART)
- ❑ **Méthodes orientées objet** (UML, OMT)
- ❑ **Méthodes formelles** (B, Z, VDM, Lotos)

Spécification fonctionnelle

Analyse Structurée (SA)

- ❑ **Démarche:** recenser les fonctionnalités à implanter
- ❑ **Résultat:** cahier des charges fonctionnel
- ❑ **On distingue**
 - ❑ **Les fonctions de service:** besoins des utilisateurs
 - ❑ **Les fonctions techniques:** requises pour implanter les fonctions de service
 - ❑ **pour chaque fonction, on précise**
 - ❑ son importance
 - ❑ des critères de qualité
- ❑ **Application: Approche fonctionnelle**
 - ❑ Les Diagrammes de Flux de Données (DFD)
 - ❑ SADT (Structured Analysis and Design Technique)

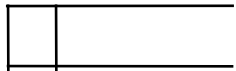
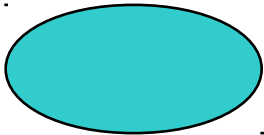
Approche fonctionnelle



Méthode des DFD

- ❑ Il s'agit d'une technique semi-formelle et opérationnelle. Les DFD décrivent des collections de données manipulées par des fonctions. Les données peuvent être persistantes (dans des stockages) ou circulantes (flots de données).
- ❑ La représentation graphique classique distingue :
 - ❑ les *fonctions*
 - ❑ les *stockages*
 - ❑ les *flots*
 - ❑ les *entités externes*

Méthode des DFD



Fonction: activité qui manipule des données

Flot de données: cheminement des données

Stockages ou dépôts : collection de données

Entité externe: source ou destination de données

•Règles de construction des DFDs

- le niveau 01 présente le système comme un seul processus:
diagramme de contexte
- les flots indiquent des transferts de données
- tous les composants des diagrammes doivent être étiquetés
- ne montrer un dépôt de données qu'à partir du moment où il est interface entre plusieurs processus
- les raffinements s'arrêtent lorsque les processus ne se décomposent plus

Les Diagrammes de Flux de Données

❑ Diagramme de contexte

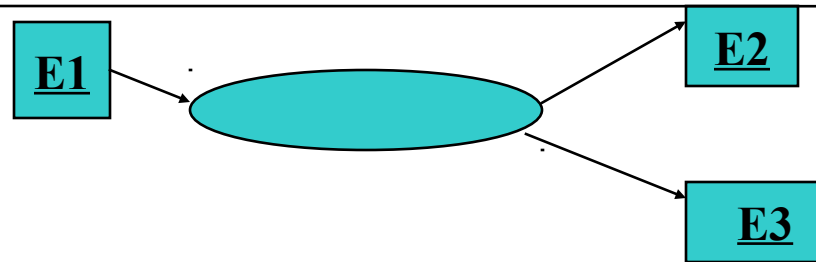
- associé à un diagramme de flot de données
- représente les échanges de flots de données avec les acteurs extérieurs du système à modéliser

❑ Diagrammes de niveau 1 et plus

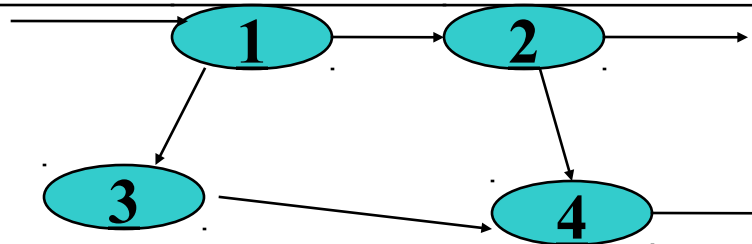
- DFD niveau 1 construit en prenant le processus représentant le système en le découpant en d'autres unités de traitement
- DFDs niveau i construits en détaillant les processus définis au niveau $i-1$ en montrant les flots de données entre ces unités

Les Diagrammes de Flux de Données

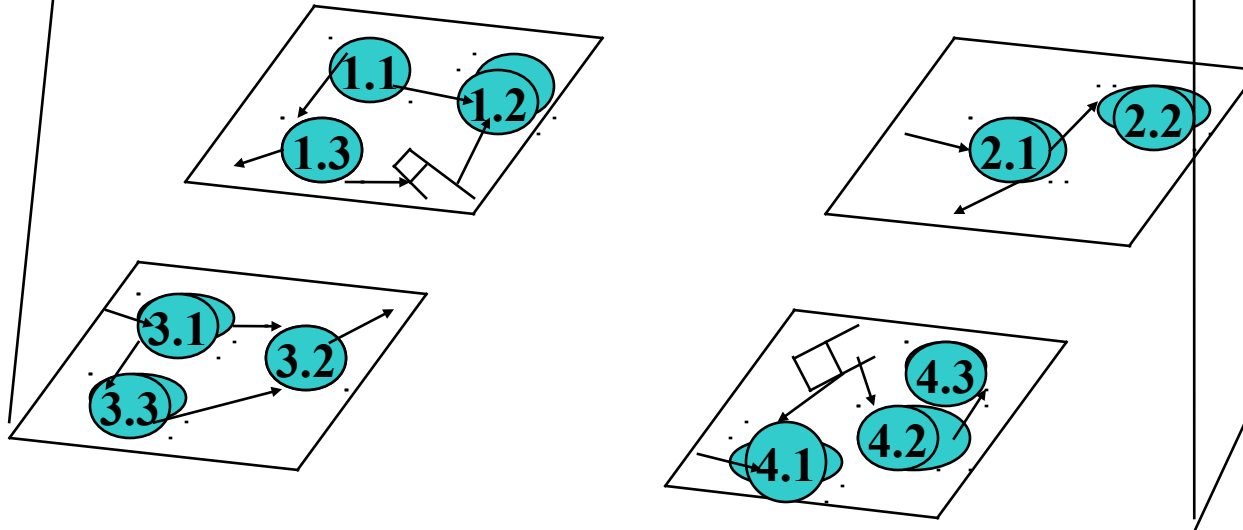
Diagramme de contexte



Niveau 1

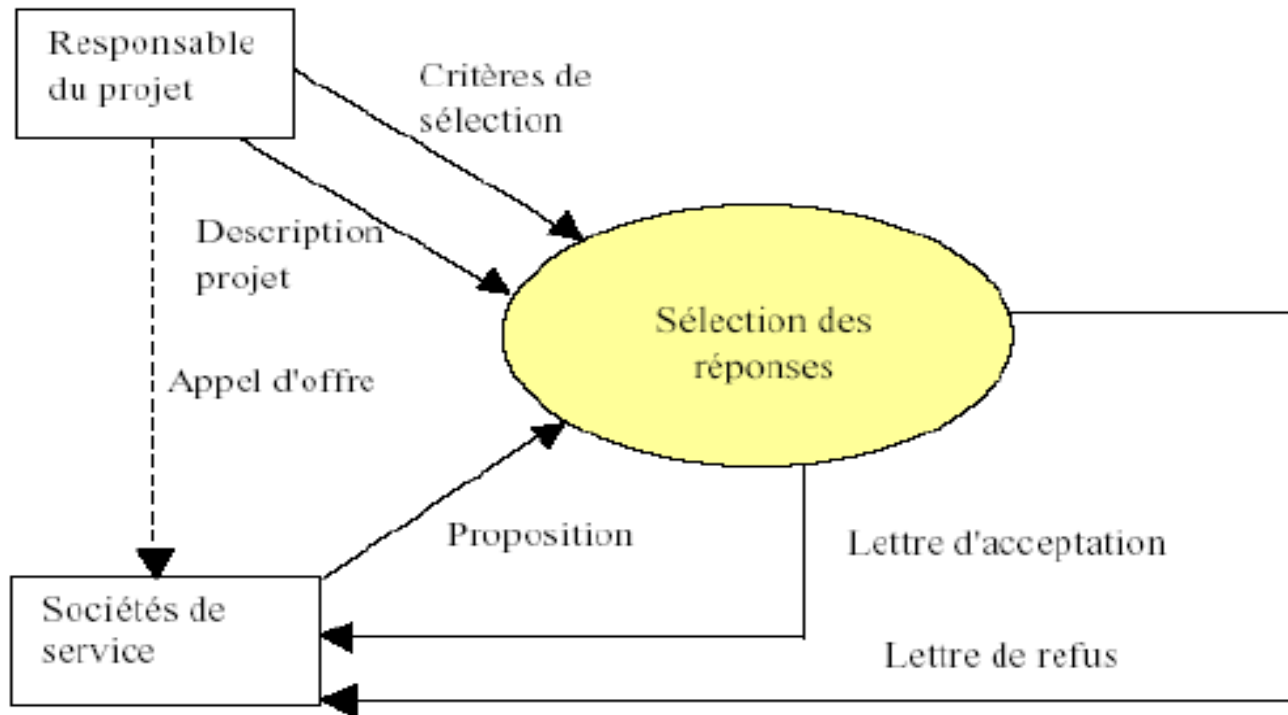


Niveau 2

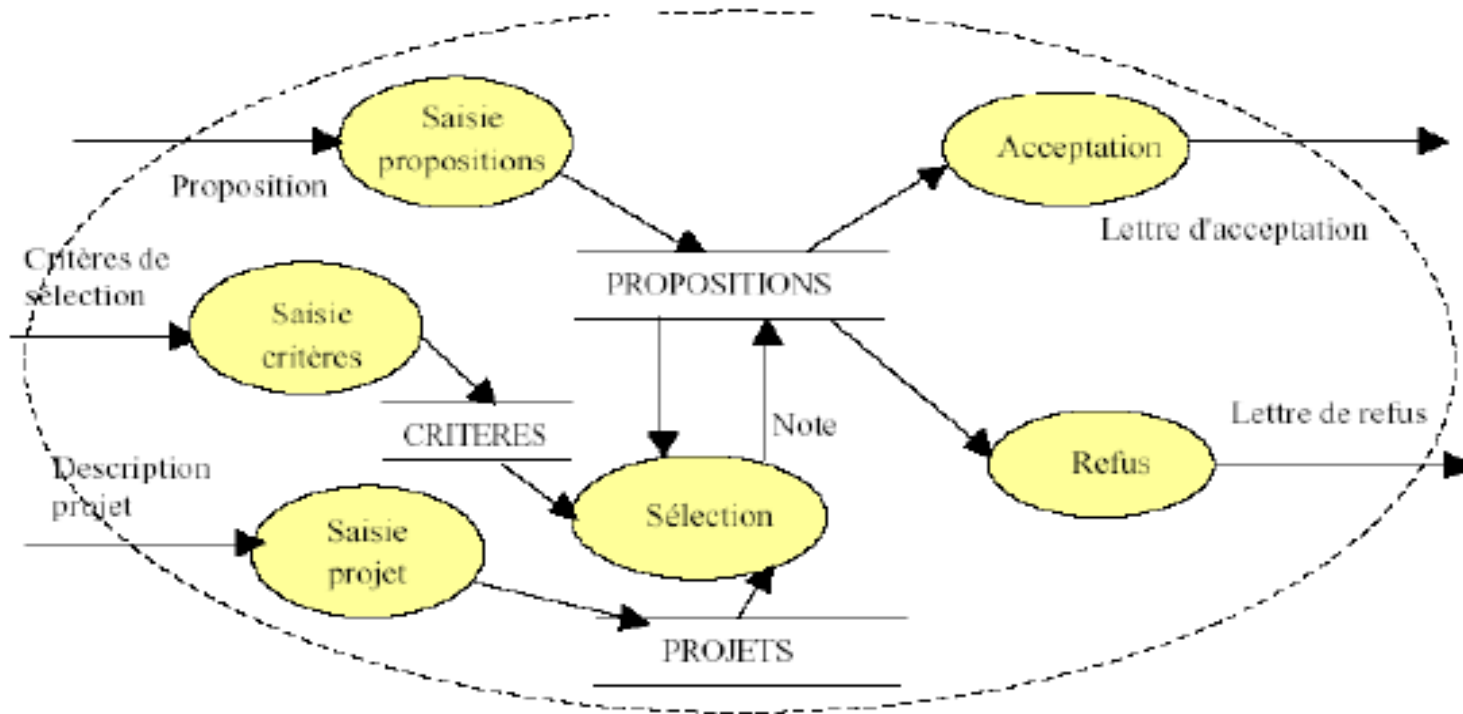


Un exemple de DFD :

le diagramme de contexte de la sélection des réponses à un appel d'offre



Raffinement du DFD précédent : la fonction « sélection des réponses » est raffinée ici.



Méthode orientée Flux de Données

❑ **Résultat de la méthode**

- ❑ ensemble de Diagrammes de Flux avec le nombre de niveaux de raffinements nécessaires
- ❑ dictionnaire des données qui documente la totalité du système tout au long du développement
- ❑ description en pseudo code des algorithmes

❑ **Avantages**

- ❑ méthode relativement formelle
- ❑ applique tous les principes d'analyse
- ❑ méthode facile à apprendre

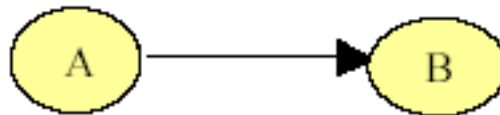
Méthode orientée Flux de Données

❑ Faiblesses

- ❑ absence d'indication du flot de contrôle
- ❑ faiblesse des outils de description de la logique des traitements
- ❑ apparition de formes 'pathologiques', comme par exemple: trou noir



- ❑ plusieurs interprétations sont possibles pour le DFD élémentaire suivant:



A produit une donnée et attend que B la traite pour en produire une autre, ou

A et B sont des processus autonomes avec un tampon entre eux

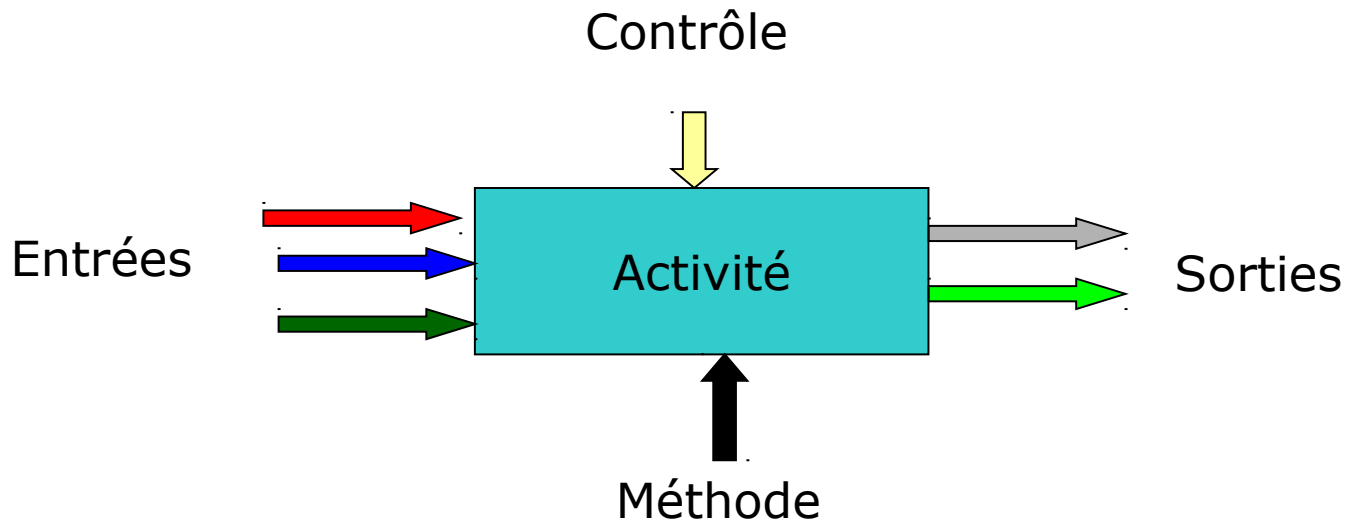
Conclusion DFD

- ❑ Pour ces raisons les DFDs sont :
 - ❑ soit complétés par d'autres spécifications
 - ❑ soit étendus
- ❑ Ils connaissent un très grand succès pour spécifier les fonctions d'un système à cause de leur *simplicité et de leur facilité de compréhension par des non informaticiens*.

Structured Analysis and Design Technique

SADT

Représentation sous forme d'actigrammes

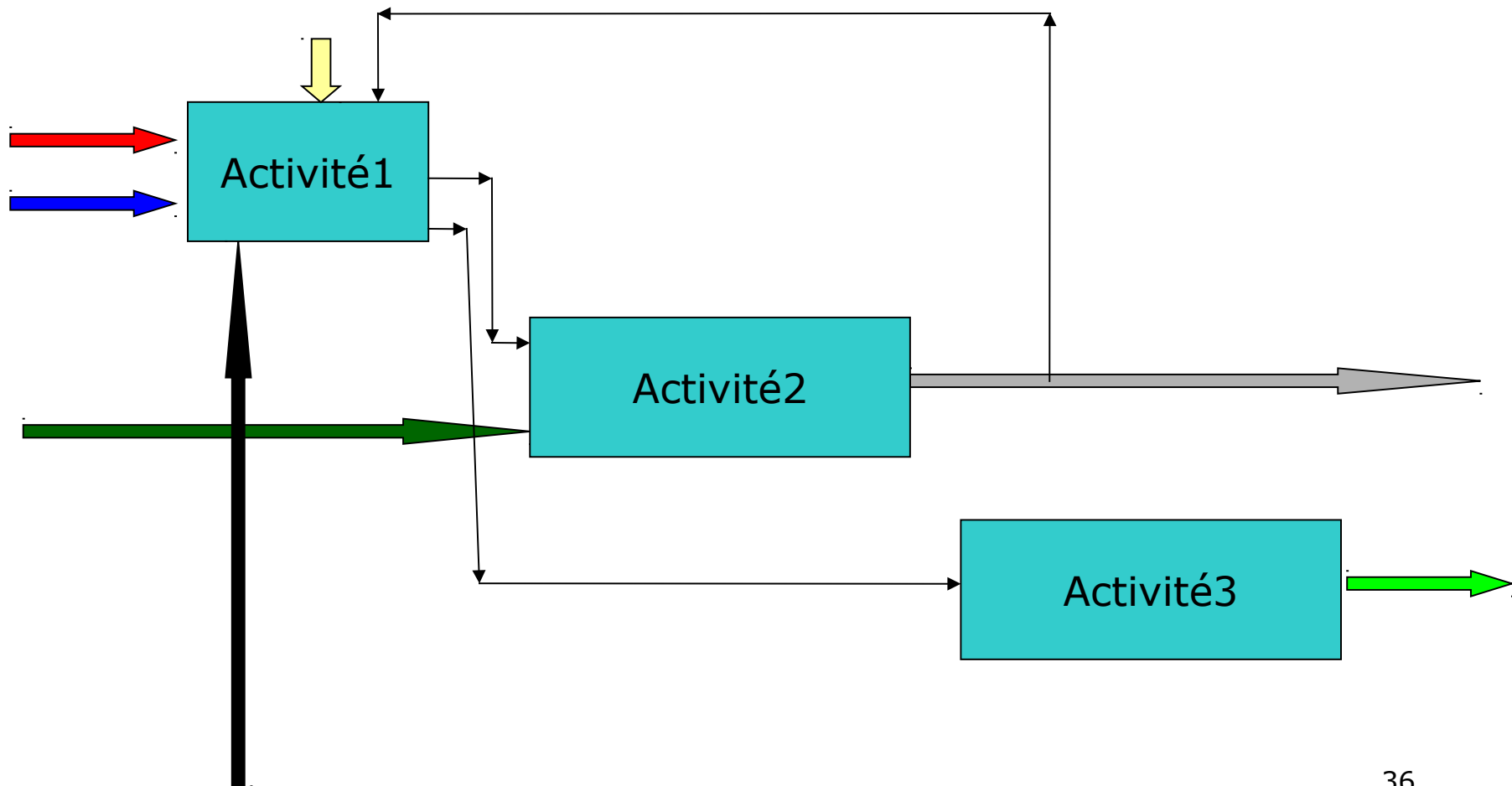


Les entrées et sorties sont des données

Structured Analysis and Design Technique

SADT

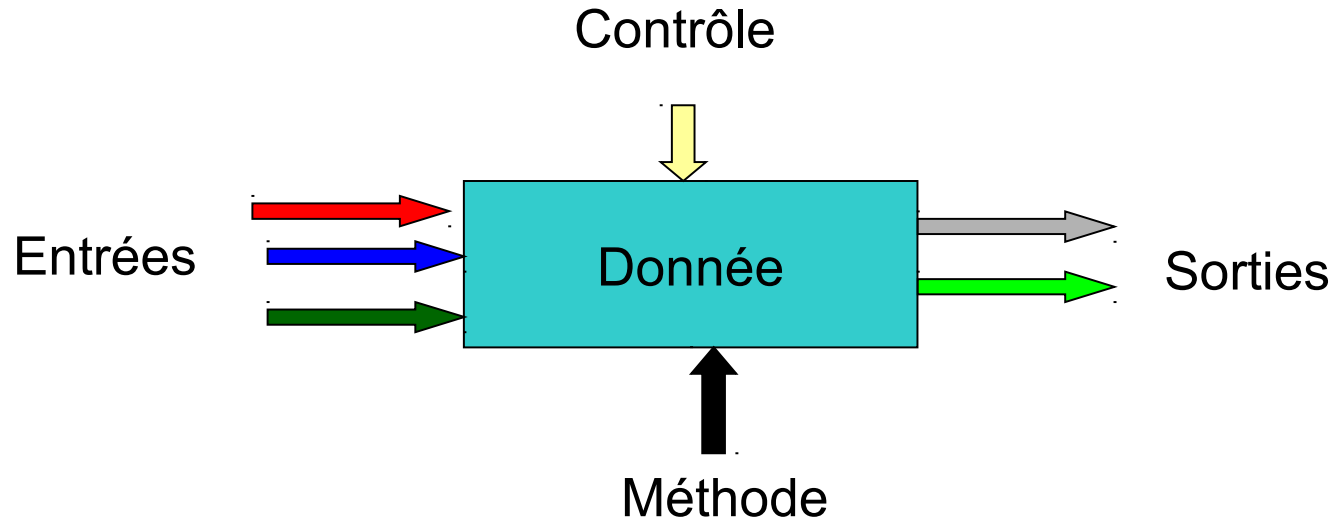
Décomposition de la boîte « Activité »



Structured Analysis and Design Technique

SADT

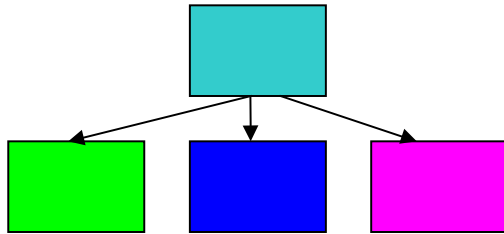
Représentation sous forme de datagrammes



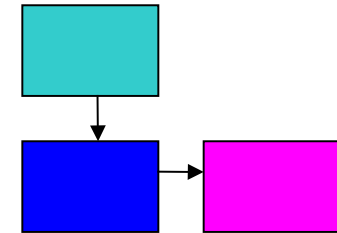
Les entrées et sorties sont des activités

Approche structurée

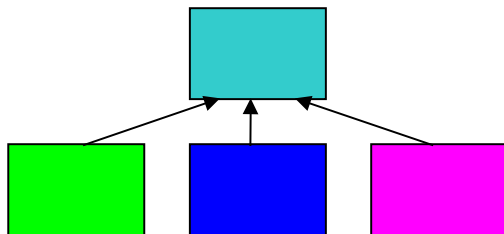
Descendante



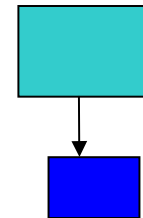
Séquentielle



Ascendante



Réursive



La Conception

Conception

Objectifs

- Connaître quelques méthodes qui permettent de développer des systèmes logiciels avec une approche structurée
- Assimiler la démarche méthodologique de conception
- Maîtriser les qualités d'une conception

Conception générale

❑ **Buts**

- ❑ mettre en place des entités de base
- ❑ faire apparaître les choix de réalisation
- ❑ définir un mode de fonctionnement général du système

❑ **Principes**

- ❑ données et fonctions, abstraction, raffinements successifs, primitives
- ❑ modularité
- ❑ indépendance fonctionnelle

❑ **Livrable conception générale**

- ❑ dossier d'architecture (fonctions logicielles, structure banque de données, interaction usager système)
- ❑ manuel utilisateur
- ❑ planning conception détaillée

Les étapes de conception

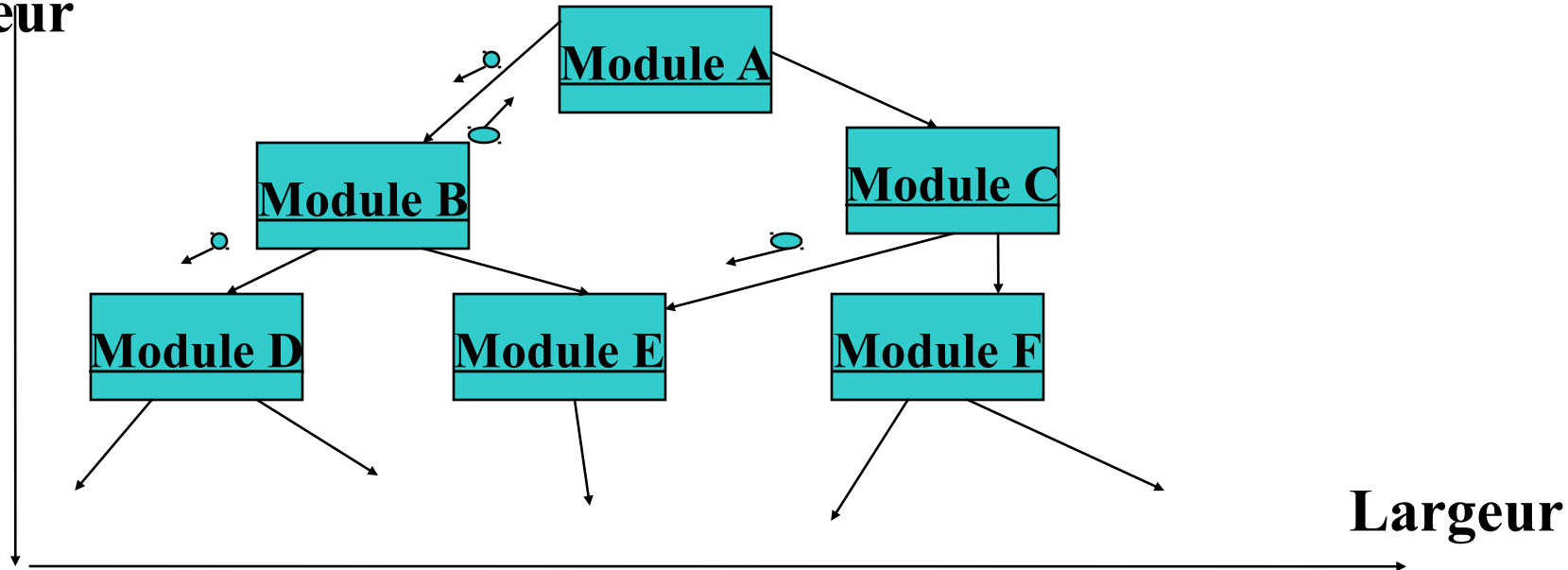
- ❑ La conception d'architecture: identification des sous-systèmes et des relations qui existent entre eux
- ❑ La spécification abstraite: spécification des sous-systèmes
- ❑ La conception d'interfaces: description des interfaces
- ❑ La conception de composants: découpage des sous-systèmes en plusieurs composants
- ❑ La conception des structures de données: définition des structures de données
- ❑ La conception d'algorithmes: conception des algorithmes pour chacune des fonctions

Outil : Diagramme de structure

- ❑ Organisation hiérarchique des différents modules du système (du général au particulier)
- ❑ Un module pourra être programmé comme un module, une procédure, une fonction ou une autre unité de traitement
- ❑ **Quatre types d'unités ou de modules:**
 - ❑ **Entrée:** unité chargée d'accepter des données des périphériques d'entrée et de les transmettre aux unités de traitement
 - ❑ **Sortie:** unité chargée d'accepter les données des unités de traitement et de les transmettre aux entités externes
 - ❑ **Transformation:** unité qui accepte une ou plusieurs données d'autres unités, les transforme, les traite et les transmet à d'autres unités
 - ❑ **Coordination:** unité responsable du contrôle et de la gestion d'autres unités

Diagramme de structure

Profondeur



Largeur

Profondeur: niveaux de raffinement

Largeur: degré de décomposition fonctionnelle

Fan-out: nombre de modules subordonnés d'un module

Fan-in: nombre de modules contrôlant un module

Conception détaillée

❑ Buts

- ❑ spécifier la manière dont chacune des entités de base définie dans la phase de conception générale sera réalisée et la manière dont ils interagiront
 - ❑ spécification des modules
 - ❑ description de la banque de données
 - ❑ description des E/S

❑ Livrable conception détaillée

- ❑ Structures de données
- ❑ Détail procédural
- ❑ Dossiers de tests d'intégration
- ❑ Dossiers de tests unitaires
- ❑ Planning de la phase de codage

Propriétés de conception

Cohésion: définit
l'homogénéité de l'intérieur
d'un module

Cohésion de

- coïncidence
- logique
- temporelle
- de communication
- séquentielle
- fonctionnelle

Couplage: définit le degré
de liaison entre modules

Couplage de

- contenu
- de COMMON
- externe
- de contrôle
- de données

Analyse et conception d'une interface

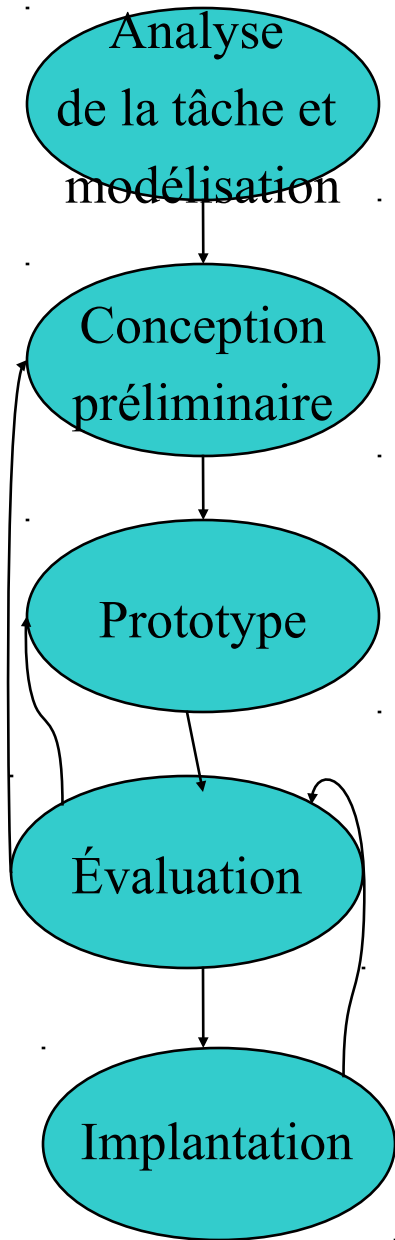
❑ **Conseils pour faire une interface usager**

- ❑ Analyse de tâches
- ❑ Conception ou choix d'une métaphore
- ❑ Choix d'un ensemble d'outils
- ❑ Conception préliminaire de l'interface
- ❑ Prototypage avec outil de haut niveau
- ❑ Cycle évaluation modification
- ❑ Implantation avec l'ensemble d'outils

❑ **Critères quantitatifs d'évaluation d'une interface usager**

- ❑ Temps d'apprentissage
- ❑ Performance
- ❑ Taux d'erreurs par les usagers
- ❑ Satisfaction subjective

Cycle de développement des interfaces



Facteurs humains: diversité des usagers

Conception préliminaire:

- Aspects visuels, choix des composants
- Disposition
- Actions