



# COURS « GÉNIE LOGICIEL 1 »

## Chapitre4: Modélisation et techniques de spécification

Niveau: II2

Enseignante: Rim DRIRA

drirarim@gmail.com

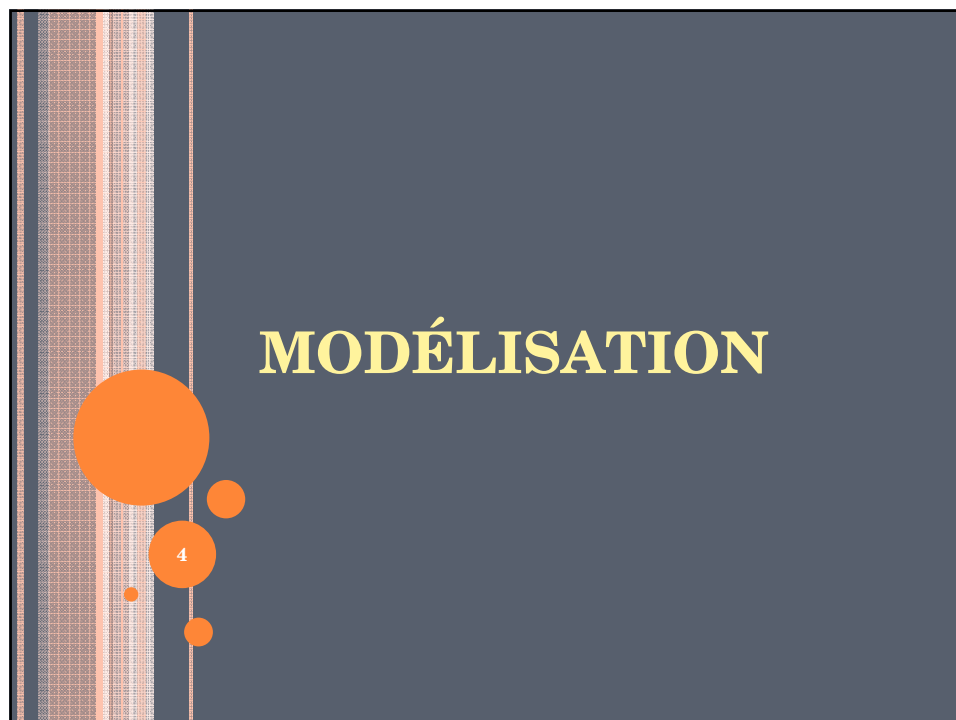
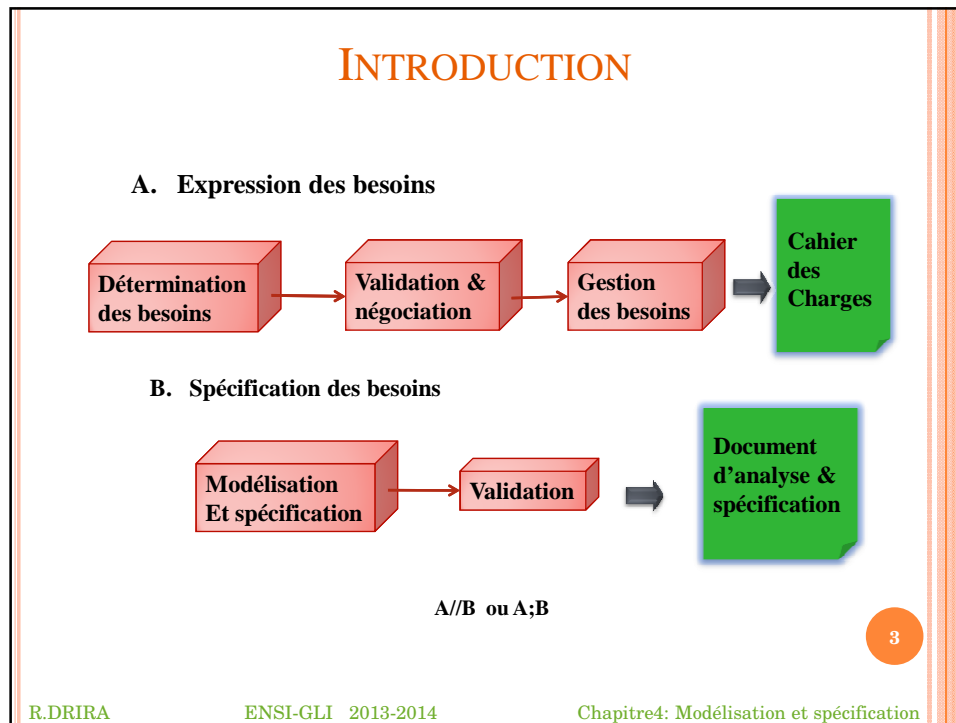
Bureau 215

AU: 2013/2014

## PLAN

1. Modélisation
2. Styles de spécification
3. Approches de spécification

2



## QU'EST CE QU'UN MODÈLE

- **Un modèle est**
  - une représentation théorique d'une réalité
- **Il a pour utilité de**
  - décrire, interpréter et prévoir des événements dans **le cadre de cette réalité**
    - **Exemple :** le modèle atomique permet de décrire des phénomènes chimiques, mais pas la gravité
  - mieux comprendre le système que l'on développe
  - Support de communication

5

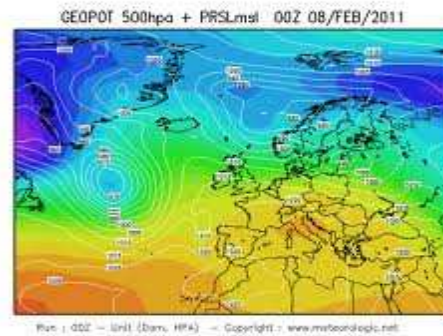
## QU'EST CE QU'UN MODÈLE

- Concrètement, un modèle permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement de manière significative.
- Il reflète ce que le concepteur **croit important** pour la compréhension et la prédiction du phénomène modélisé.
- Les limites du phénomène modélisé **dépendent des objectifs du modèle.**

6

## EXEMPLE 1

### □ Le modèle météorologique



À partir de données d'observation (satellite, ...), il permet de prévoir les conditions climatiques pour les jours à venir.

7

## EXEMPLE 2



### Le plan d'un bâtiment

- Les plans sont des modèles qui donnent une vue d'ensemble du système concerné.
- Pour la construction d'un immeuble, il faut préalablement élaborer de nombreux plans :
  - plans généraux du bâtiment et de sa structure ;
  - plans d'implantation du bâtiment dans son environnement ;
  - plans détaillées des différents locaux, bureaux, appartements, etc.
  - plans des câblages électriques ;
  - plans d'écoulements des eaux, etc.
- Ce modèle, plus conceptuel, possède **différents points de vues** comme la plupart des **modèles en génie logiciel**.

8

## LES QUATRE PRINCIPES DE MODÉLISATION

1. Le **choix des modèles** à créer a une forte influence sur la manière d'aborder un problème et sur la nature de sa solution.
2. Tous les modèles peuvent avoir différents **niveaux de précision**.
3. Les meilleurs modèles ne perdent pas **le sens de la réalité**.
4. Parce qu'aucun modèle n'est suffisant à lui seul, il est préférable de **décomposer** un système en un ensemble de **petits modèles** presque indépendants.

9

## EN GÉNIE LOGICIEL

- Dans le domaine de l'ingénierie du logiciel,
  - le modèle permet de mieux répartir les tâches et d'automatiser certaines d'entre elles;
  - Le modèle est un facteur de réduction des coûts et des délais.
  - Le modèle est enfin indispensable pour assurer un bon niveau de qualité et une maintenance efficace.

10

## USAGE DES MODÈLES

- **Au niveau capture des besoins**
  - le modèle trace les frontières fonctionnelles du système;
- **Au niveau de l'analyse**
  - le modèle est une abstraction des concepts manipulés par les utilisateurs au point de vue statique et dynamique;
- **Au niveau de la conception**
  - le modèle représente les concepts utilisés par les outils, les langages et les plates-formes ;
- **Au niveau de déploiement**
  - le modèle représente les matériels et les composants logiciels à interconnecter;

11

## STYLES DE SPÉCIFICATION

12

## LES STYLES DE SPÉCIFICATION

- Le degré de formalisation :
  - **informel** : basée sur le langage naturel (libre, formaté...), croquis, ...
  - **semi-formel** : basée sur un langage structuré (graphique et/ou textuel) dont la sémantique est faible
  - **formel** : basée sur un langage formel dont le vocabulaire, la syntaxe et la sémantique sont formels

13

## SPÉCIFICATION INFORMELLE

- **Exemple :**
  - La vérification de la validité de la carte consiste à vérifier que la carte introduite par un utilisateur provient d'une banque reconnue, qu'elle est à jour, et qu'elle contient des informations appropriées ainsi que des détails sur les dates et les montants des précédents retraits.

14

## SPÉCIFICATION INFORMELLE

### ○ **Avantage**

- liberté pour l'auteur

### ○ **Inconvénients**

- *Ambiguïté* : un même mot ne fait pas toujours référence au même concept chez deux locuteurs différents ou dans deux contextes différents.
- *Flexibilité excessive* : un même concept peut être expliqué de plusieurs façons différentes, ce qui complique la recherche d'information.
- *Modularisation difficile* : le partage des mots entre différents modules rend difficile la traçabilité des concepts (quel module est en charge de quoi ?).

15

## SPÉCIFICATION SEMI-FORMELLE

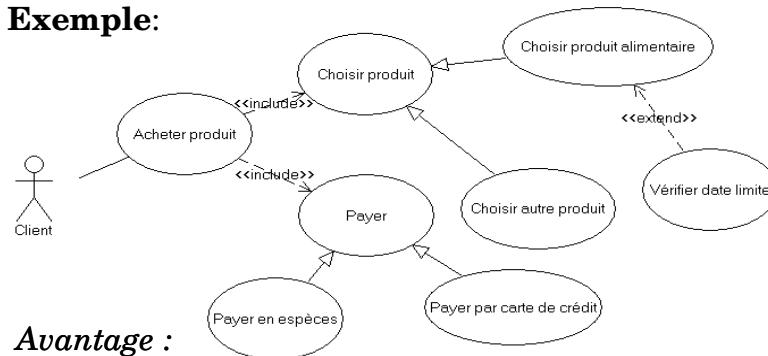
- Se base sur une notation graphique :
  - Introduit un aspect formel mais diagrammes annotés généralement par du texte informel (pour cette raison, on dit semi-formelle).
- Ces notations sont particulièrement pratiques pour fournir une vue d'ensemble, statique ou dynamique, d'un système ou d'un sous-système.
- Cependant, **leur sémantique doit être précisée.**

16



## SPÉCIFICATION SEMI-FORMELLE

### Exemple:



#### ■ *Avantage :*

- ◆ Compromis lisibilité-formalisme
- ◆ Support de communication

#### ■ *Inconvénient :*

- ◆ Sémantique faible

17

R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## SPÉCIFICATION FORMELLE

- Une spécification d'un logiciel est formelle si elle est exprimée avec un langage qui possède:
  - un vocabulaire et une syntaxe formellement définis;
  - une sémantique basée sur les mathématiques.

18

R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## EXEMPLE: RÉSULTATS D'UNE COURSE

```

MACHINE
  Résultats
SETS COUREUR
VARIABLES classement
INVARIANT classement ∈ iseq(COUREUR)
INITIALISATION classement := []
OPERATIONS
arrivee(rr) = PRE rr ∈ COUREUR ∧ rr ∉ ran(classement)
THEN classement := classement ← rr
END;
rr <-- place(pp) =
PRE pp ∈ NAT1 ∧ pp ≤ size(classement)
THEN rr := classement(pp)
END;
disqualifié(pp) =
PRE pp ∈ NAT1 ∧ pp ≤ size(classement)
THEN classement := (classement ↑ pp-1) ∩ (classement ↓ pp)
END;
ss <-- médaillés = BEGIN ss := classement ↑ 3 END;
pp <-- position(rr) =
PRE rr ∈ COUREUR ∧ rr ∈ ran(classement) THEN pp := classement-1(rr) END
END

```

19

## SPÉCIFICATION FORMELLE

### ○ Avantages :

- Améliorer la qualité du logiciel;
- Rigueur et précision des spécifications;
- Automatisation

### ○ Inconvénients :

- Nécessite une certaine qualification du client, utilisateurs et développeurs
- Ne facilite pas la communication avec les utilisateurs;

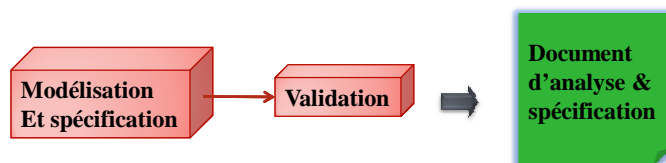
20

## APPROCHES (FONCTIONNELLE, OBJET)

21

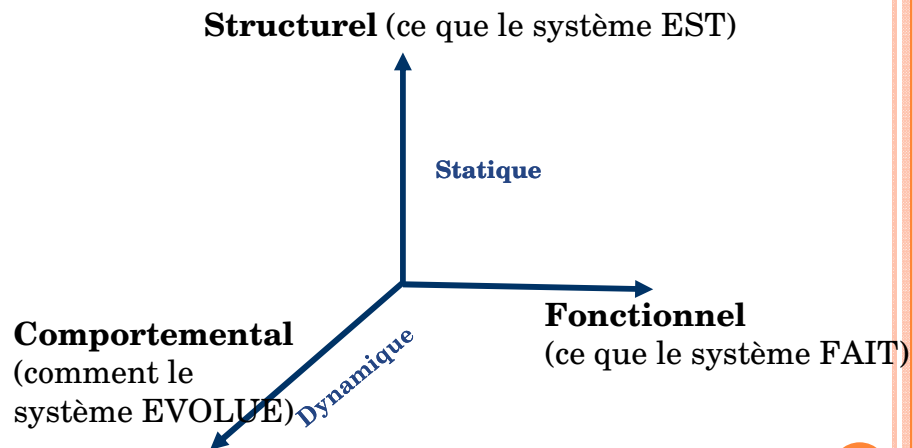
### SPÉCIFICATION

- La spécification aura pour but de décrire avec rigueur :
  - La structure du système (**vue structurelle**)
  - Les fonctions du systèmes (**vue fonctionnelle**)
  - Les changements d'états et le contrôle du système (**vue comportementale**)



22

## AXES DE MODÉLISATION D'UN SYSTÈME



23

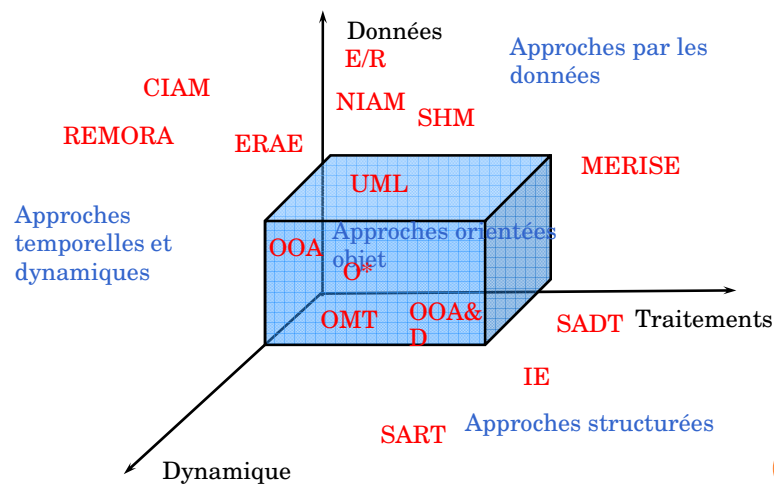
R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## LE JUNGLE DES MODÈLES

Trois perspectives : Données, Traitements, Dynamique



24

R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

### APPROCHE FONCTIONNELLE: VUE FONCTIONNELLE

- **Modélisation à l'aide de diagrammes de flot de données DFD (data flow diagramm)**
- Le DFD représente ce que fait le système sous forme d'un réseau de fonctions transformant des flux.
- Représentation graphique du cheminement des données à travers un système
- les composants logiciels sont vus comme des fonctions qui transforment des données d'entrée en données de sortie.
- **Intérêt:** Décrit le système dans son environnement; Base de référence pour les étapes ultérieures et vérification-validation.

25

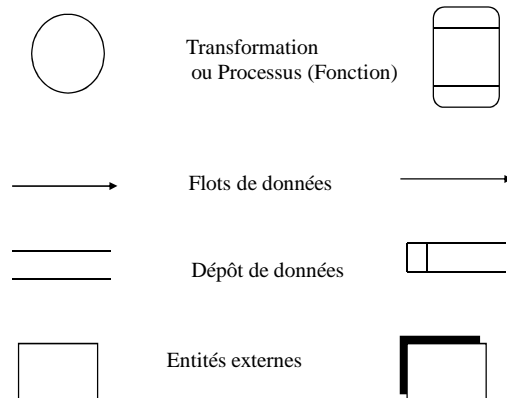
### MODÈLE FONCTIONNEL -DFD

- Données organisées de différentes façons:
  - **Stockées dans des dépôts**
  - **Évoluant dans des flots**
  - **Transférées de/vers l'extérieur**

26

## MODÈLE FONCTIONNEL -DFD

- Plusieurs variantes de notation (Gane & Sarson, Yourdon, SSADM)



27

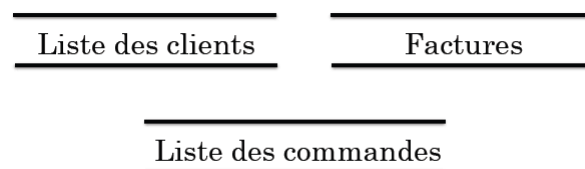
R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## DFD - DÉPÔT DE DONNÉES

- Stockage de données.
- Aucune matérialisation particulière prescrite.
- Identifié par un nom unique.



28

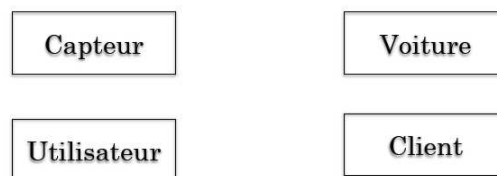
R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## DFD - ENTITÉS EXTERNES

- Externes au système.
- Sources et destinations de données.
- Fournisseurs et consommateurs de données.
- Entités indépendantes du processus.
- Synonymes: agents, acteurs...



29

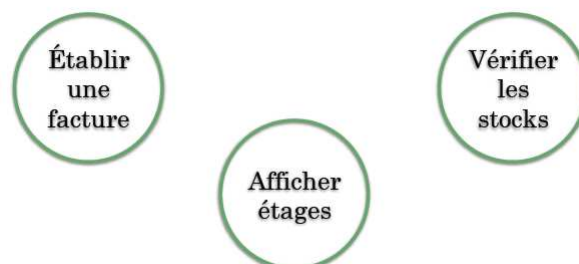
R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## DFD- TRANSFORMATION

- Représente une fonction du système.
- Transforme les flux entrants en flux sortants.
- Est identifié par un nom unique (verbe).
- La présence d'un traitement signifie automatiquement qu'une transformation a lieu sur des données
- Remarque: Numéroté les traitements



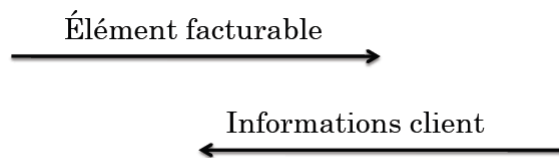
30

R.DRIRA

on et spécification

## DFD – FLUX DE DONNÉES

- Liant les processus entre eux.
- Liant les processus aux entités externes.
- Liant les processus aux dépôts de données.
- Une flèche indique le sens du flux.
- Identifié par un nom unique.



31

## DFD – FLUX DE DONNÉES

- **Règles de construction des flux de données :**
  - R1: Il faut attribuer un nom à chaque flux de données (une exception est possible lorsqu'un flux entre ou sort d'un dépôt de données).
  - R2: Lorsqu'un flux entre dans un traitement et autre en sort, les deux flux doivent porter des noms différents pour refléter la transformation.
  - R3: La source ou la destination d'un flux doit être un processus.

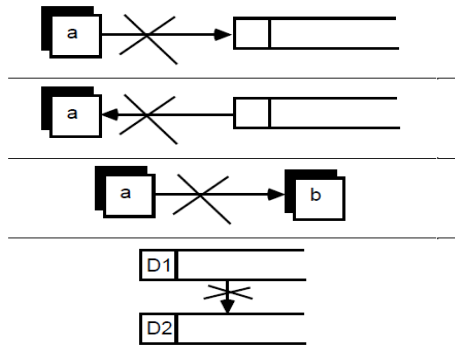
32



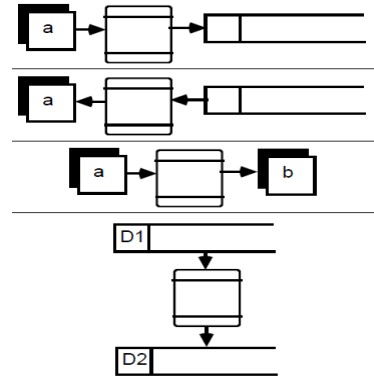
## DFD – FLUX DE DONNÉES

### ○ Illustration pour R3

**Non**



**Oui**



R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## DFD - NIVEAUX D'ABSTRACTION

- **DFD peut être utilisé pour représenter le système à différents niveaux d'abstraction:**
- **Le niveau d'abstraction le plus élevé** (niveau 0) est appelé **diagramme de contexte**:
  - Un seul processus représente tout le système sans aucun détail
  - Il décrit le système dans son environnement en illustrant toutes les sources, les flux entrants, les flux sortants et les destinations.
- **Des DFD de niveau d'abstraction plus bas:** Une transformation peut être « éclatée » et donne naissance à un nouveau DFD qui montre le traitement des données à un niveau plus détaillé.
- Cette décomposition descendante (Top-Down) peut continuer jusqu'à avoir une fonction par processus
- Les données sont aussi raffinées à travers les processus

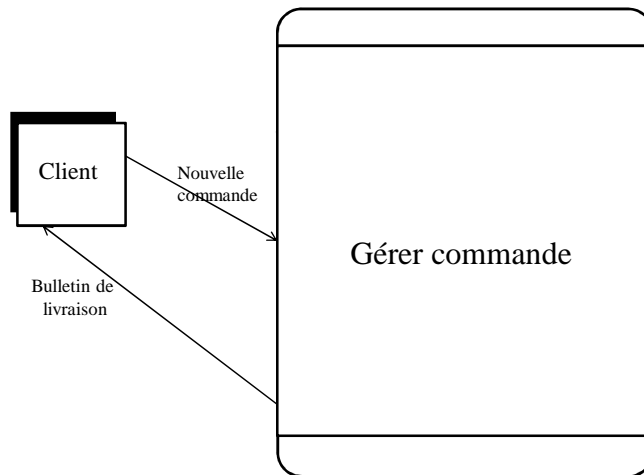
34

R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## EXEMPLE 1- DFD 0



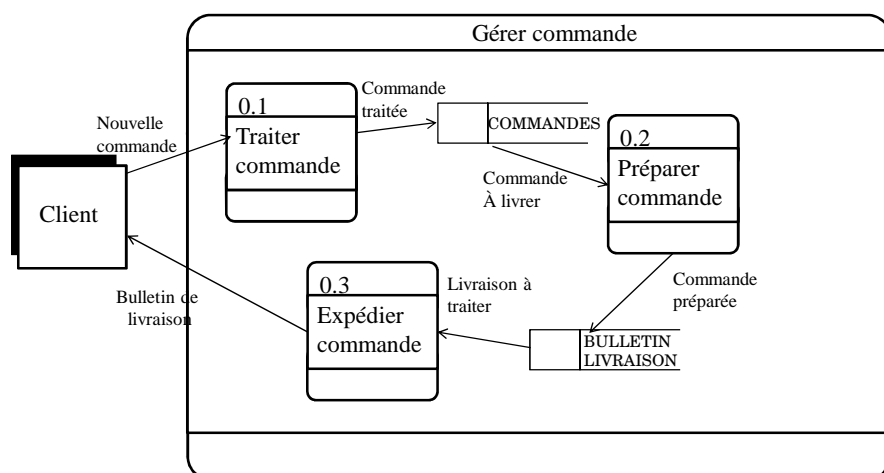
35

R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## EXEMPLE 1- DFD 1



36

R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## EXEMPLE2

- Le système sera installé dans l'unité de soins intensifs de l'hôpital. Cette unité comporte 8 lits destinés à recevoir des patients dont l'état est critique et dont les signes vitaux doivent être surveillés 24/24.
- Les signes vitaux à surveiller sont l'électrocardiogramme, le rythme respiratoire et la pression artérielle.
- La surveillance consiste à saisir les signes vitaux, à les afficher sur des moniteurs aux lits et au poste des infirmières, à les comparer avec une banque de référence et à émettre éventuellement des alarmes.
- Le système doit produire des rapports sur chaque patient.

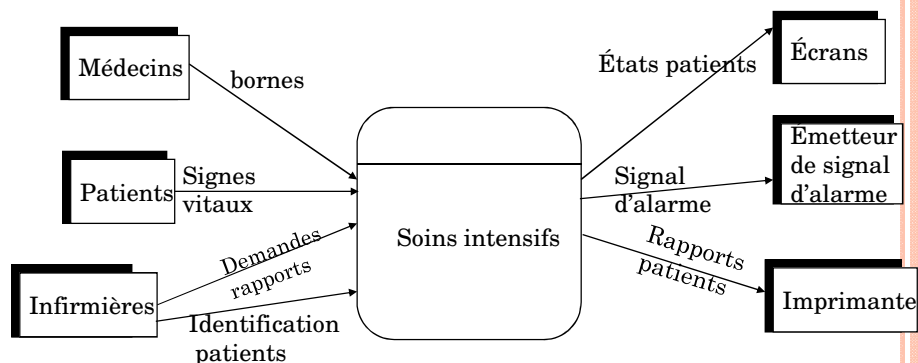
37

R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## EXEMPLE2 - DIAGRAMME DE CONTEXTE (DFD 0)



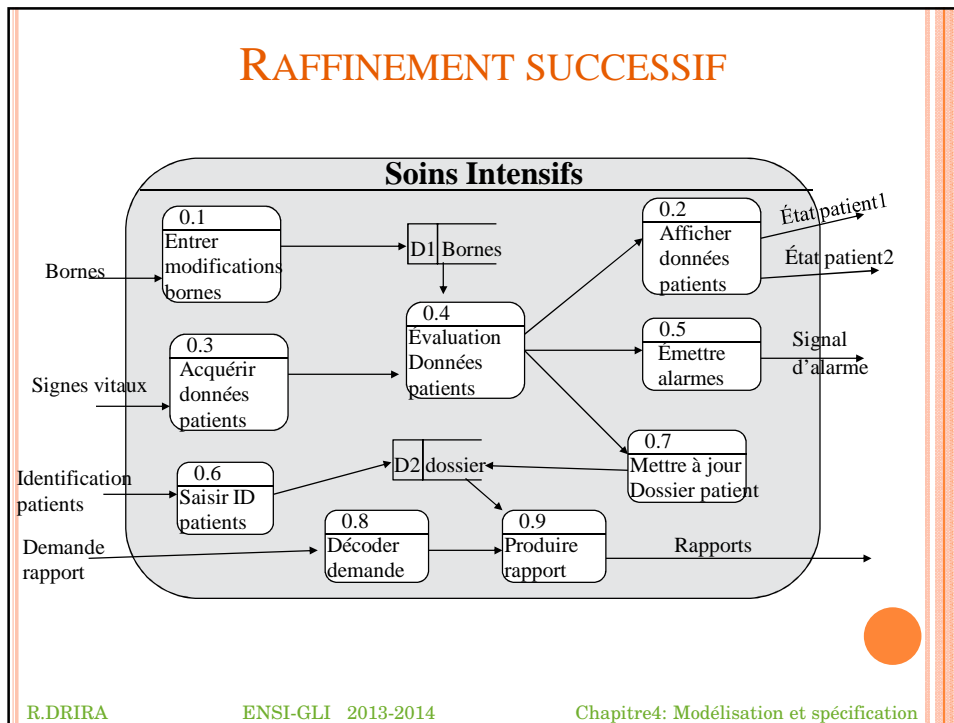
38

R.DRIRA

ENSI-GLI 2013-2014

Chapitre4: Modélisation et spécification

## RAFFINEMENT SUCCESSIF



## RAFFINEMENT SUCCESSIF

