

Introduction à l'analyse de systèmes d'information

© Olivier Caron

Les Objectifs de l'analyse

- ✓ Description d'un problème :
 - ▶ Un système d'information à informatiser

Les Objectifs de l'analyse

- ✓ Description d'un problème :
 - ▶ Un système d'information à informatiser
 - ▶ Un système d'information déjà informatisé à faire évoluer.
- ✓ Comment est décrit le problème ?
 - ▶ En langage naturel (non informaticiens)

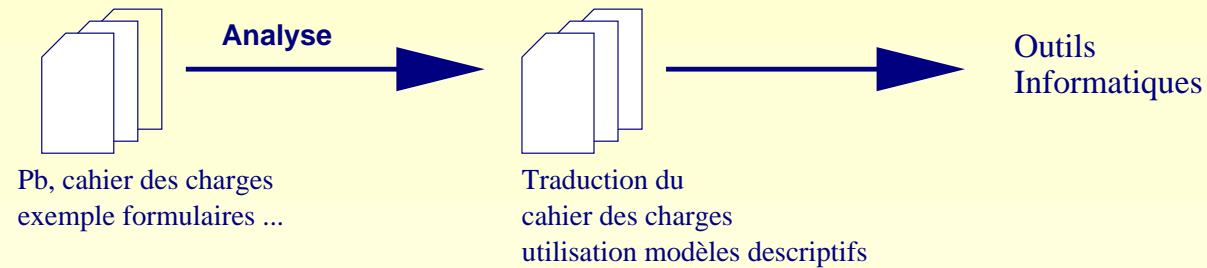
Les Objectifs de l'analyse

- ✓ Description d'un problème :
 - ▶ Un système d'information à informatiser
 - ▶ Un système d'information déjà informatisé à faire évoluer.
- ✓ Comment est décrit le problème ?
 - ▶ En langage naturel (non informaticiens)
 - ▶ Sous forme de feuilles d'état : factures, listings, formulaires, . . . qu'on dispose déjà où qu'on voudrait avoir.

Les Objectifs de l'analyse

- ✓ Description d'un problème :
 - ▶ Un système d'information à informatiser
 - ▶ Un système d'information déjà informatisé à faire évoluer.
- ✓ Comment est décrit le problème ?
 - ▶ En langage naturel (non informaticiens)
 - ▶ Sous forme de feuilles d'état : factures, listings, formulaires, . . .
qu'on dispose déjà où qu'on voudrait avoir.

Les étapes de l'analyse



La phase d'analyse consiste à utiliser **une méthode** qui, à partir d'une description non formelle d'un système d'information, va transcrire, dans un premier temps, ce système en utilisant **un modèle plus formel**, puis traduire ce schéma en *fichiers, base de données, programmes, . . .*

La phase de traduction

- ✓ La phase de traduction permet :
 - ▶ de bien comprendre les besoins des utilisateurs

La phase de traduction

- ✓ La phase de traduction permet :
 - ▶ de bien comprendre les besoins des utilisateurs
 - ▶ d'évaluer si le cahier des charges est réalisable

La phase de traduction

- ✓ La phase de traduction permet :
 - ▶ de bien comprendre les besoins des utilisateurs
 - ▶ d'évaluer si le cahier des charges est réalisable
 - ▶ de valider le modèle de données créé.

La phase de traduction

- ✓ La phase de traduction permet :
 - ▶ de bien comprendre les besoins des utilisateurs
 - ▶ d'évaluer si le cahier des charges est réalisable
 - ▶ de valider le modèle de données créé.
 - ▶ de disposer d'outils automatique de traduction vers des programmes, base de données.

Les composantes d'un système

- ✓ Selon la complexité des systèmes, 4 composantes sont à prendre en compte :
 - ▶ Les données (papiers, documentation, fichiers, . . .)

Les composantes d'un système

- ✓ Selon la complexité des systèmes, 4 composantes sont à prendre en compte :
 - ▶ Les données (papiers, documentation, fichiers, . . .)
 - ▶ Les traitements (consultation, recherche, mise à jour, inscription)

Les composantes d'un système

- ✓ Selon la complexité des systèmes, 4 composantes sont à prendre en compte :
 - ▶ Les données (papiers, documentation, fichiers, . . .)
 - ▶ Les traitements (consultation, recherche, mise à jour, inscription)
 - ▶ Les processeurs (les machines qui effectuent des traitements)

Les composantes d'un système

- ✓ Selon la complexité des systèmes, 4 composantes sont à prendre en compte :
- ▶ Les données (papiers, documentation, fichiers, . . .)
 - ▶ Les traitements (consultation, recherche, mise à jour, inscription)
 - ▶ Les processeurs (les machines qui effectuent des traitements)
 - ▶ Les hommes (les acteurs, les utilisateurs, les administrateurs)

Les composantes d'un système

- ✓ Selon la complexité des systèmes, 4 composantes sont à prendre en compte :
 - ▶ Les données (papiers, documentation, fichiers, . . .)
 - ▶ Les traitements (consultation, recherche, mise à jour, inscription)
 - ▶ Les processeurs (les machines qui effectuent des traitements)
 - ▶ Les hommes (les acteurs, les utilisateurs, les administrateurs)
- ✓ Deux approches co-existent :
 - ▶ L'approche orientée données (statique)

Les composantes d'un système

- ✓ Selon la complexité des systèmes, 4 composantes sont à prendre en compte :
 - ▶ Les données (papiers, documentation, fichiers, . . .)
 - ▶ Les traitements (consultation, recherche, mise à jour, inscription)
 - ▶ Les processeurs (les machines qui effectuent des traitements)
 - ▶ Les hommes (les acteurs, les utilisateurs, les administrateurs)
- ✓ Deux approches co-existent :
 - ▶ L'approche orientée données (statique)
 - ▶ L'approche orientée traitement (dynamique)

Les composantes d'un système

- ✓ Selon la complexité des systèmes, 4 composantes sont à prendre en compte :
 - ▶ Les données (papiers, documentation, fichiers, . . .)
 - ▶ Les traitements (consultation, recherche, mise à jour, inscription)
 - ▶ Les processeurs (les machines qui effectuent des traitements)
 - ▶ Les hommes (les acteurs, les utilisateurs, les administrateurs)
- ✓ Deux approches co-existent :
 - ▶ L'approche orientée données (statique)
 - ▶ L'approche orientée traitement (dynamique)
 - ▶ L'approche objet

La méthodologie MERISE

- ✓ **Méthodologie** largement répandue en France
- ✓ Séparation de l'analyse des données de celle des traitements

La méthodologie MERISE

- ✓ **Méthodologie** largement répandue en France
- ✓ Séparation de l'analyse des données de celle des traitements
- ✓ Pour chaque approche, un modèle de conception existe :
 - ▶ Modèle conceptuel de données (modèle de type Entité-Association)

La méthodologie MERISE

- ✓ **Méthodologie** largement répandue en France
- ✓ Séparation de l'analyse des données de celle des traitements
- ✓ Pour chaque approche, un modèle de conception existe :
 - ▶ Modèle conceptuel de données (modèle de type Entité-Association)
 - ▶ Modèle de traitement, diagramme de flux d'acteurs

La méthodologie MERISE

- ✓ **Méthodologie** largement répandue en France
- ✓ Séparation de l'analyse des données de celle des traitements
- ✓ Pour chaque approche, un modèle de conception existe :
 - ▶ Modèle conceptuel de données (modèle de type Entité-Association)
 - ▶ Modèle de traitement, diagramme de flux d'acteurs
- ✓ Merise est particulièrement adaptée pour les outils informatiques tels que les syst. de gestion de fichiers et les bases de données relationnelles

La méthodologie MERISE

- ✓ **Méthodologie** largement répandue en France
- ✓ Séparation de l'analyse des données de celle des traitements
- ✓ Pour chaque approche, un modèle de conception existe :
 - ▶ Modèle conceptuel de données (modèle de type Entité-Association)
 - ▶ Modèle de traitement, diagramme de flux d'acteurs
- ✓ Merise est particulièrement adaptée pour les outils informatiques tels que les syst. de gestion de fichiers et les bases de données relationnelles
- ✓ **Les défauts de MERISE :**
 - ▶ Peu ou pas connue en dehors de l'hexagone

La méthodologie MERISE

- ✓ **Méthodologie** largement répandue en France
- ✓ Séparation de l'analyse des données de celle des traitements
- ✓ Pour chaque approche, un modèle de conception existe :
 - ▶ Modèle conceptuel de données (modèle de type Entité-Association)
 - ▶ Modèle de traitement, diagramme de flux d'acteurs
- ✓ Merise est particulièrement adaptée pour les outils informatiques tels que les syst. de gestion de fichiers et les bases de données relationnelles
- ✓ **Les défauts de MERISE :**
 - ▶ Peu ou pas connue en dehors de l'hexagone
 - ▶ Mal adaptée aux nouveaux outils : SGBDOR et SGBDOO.

Comparaison UML et MERISE

✓ UML : Unified Modeling Language (<http://www.omg.org>)

Comparaison UML et MERISE

- ✓ UML : Unified Modeling Language (<http://www.omg.org>)
- ✓ UML n'est pas une méthodologie mais une notation graphique.

Comparaison UML et MERISE

- ✓ UML : Unified Modeling Language (<http://www.omg.org>)
- ✓ UML n'est pas une méthodologie mais une notation graphique.
- ✓ UML intègre l'objet et est donc plus adaptée aux SGBDOO

Comparaison UML et MERISE

- ✓ UML : Unified Modeling Language (<http://www.omg.org>)
- ✓ UML n'est pas une méthodologie mais une notation graphique.
- ✓ UML intègre l'objet et est donc plus adaptée aux SGBDOO
- ✓ UML est adaptée à la programmation orienté-objet

Comparaison UML et MERISE

- ✓ UML : Unified Modeling Language (<http://www.omg.org>)
- ✓ UML n'est pas une méthodologie mais une notation graphique.
- ✓ UML intègre l'objet et est donc plus adaptée aux SGBDOO
- ✓ UML est adaptée à la programmation orienté-objet
- ✓ Les deux modèles sont basés sur le modèle entité-association.

Comparaison UML et MERISE

- ✓ UML : Unified Modeling Language (<http://www.omg.org>)
- ✓ UML n'est pas une méthodologie mais une notation graphique.
- ✓ UML intègre l'objet et est donc plus adaptée aux SGBDOO
- ✓ UML est adaptée à la programmation orienté-objet
- ✓ Les deux modèles sont basés sur le modèle entité-association.
- ✓ Quelques outils :
 - ▶ AMC-Designor (commercial, MERISE) : génération de tables SQL.
 - ▶ ArgoUML (gratuit, UML) : génération code Java
 - ▶ Together (gratuit, commercial, UML) : génération code C++ et Java
 - ▶ Rational Rose (commercial, UML) : tables, C++, Java, . . .

Phase I : identification des informations

- ✓ Obtenues à partir des documents fournis, une information sera décrite par :
 - ▶ Un nom : (n° client, n° facture, nomClient, . . .)
 - ▶ Son type (numérique (entier, réel, date) ou alphanumérique)

Phase I : identification des informations

- ✓ Obtenues à partir des documents fournis, une information sera décrite par :
 - ▶ Un nom : (n° client, n° facture, nomClient, . . .)
 - ▶ Son type (numérique (entier, réel, date) ou alphanumérique)
 - ▶ Sa longueur

Phase I : identification des informations

- ✓ Obtenues à partir des documents fournis, une information sera décrite par :
 - ▶ Un nom : (n° client, n° facture, nomClient, . . .)
 - ▶ Son type (numérique (entier, réel, date) ou alphanumérique)
 - ▶ Sa longueur
 - ▶ Son origine (quel document(s) ?)

Phase I : identification des informations

- ✓ Obtenues à partir des documents fournis, une information sera décrite par :
- ▶ Un nom : (n° client, n° facture, nomClient, . . .)
 - ▶ Son type (numérique (entier, réel, date) ou alphanumérique)
 - ▶ Sa longueur
 - ▶ Son origine (quel document(s) ?)
 - ▶ Des observations :
 - Des contraintes d'intégrité :
exemples : salaire > smic, age > 0
 - Des règles de calcul :
exemple : nouvelle valeur du salaire > ancienne valeur

Phase I : identification des informations (suite)

✓ Description d'une information (suite) :

▶ Sa nature : donnée calculée (C) ou non calculée (NC)

ex : passer une commande :

$\text{prix (NC)} * \text{quantité (NC)} = \text{prixCommande(C)}$

remarque : certaines informations calculées nécessitent d'être conservées, on parle d'informations calculées et mémorisées (CM)

✓ Vérification :

▶ absence de **polysème** : deux noms sont polysèmes si elles ont le même nom et expriment deux valeurs différentes.

exemple : date pour dateFacturation et pour dateCommande

Phase I : identification des informations (suite)

✓ Description d'une information (suite) :

▶ Sa nature : donnée calculée (C) ou non calculée (NC)

ex : passer une commande :

$\text{prix (NC)} * \text{quantité (NC)} = \text{prixCommande(C)}$

remarque : certaines informations calculées nécessitent d'être conservées, on parle d'informations calculées et mémorisées (CM)

✓ Vérification :

▶ absence de **polysème** : deux noms sont polysèmes si elles ont le même nom et expriment deux valeurs différentes.

exemple : date pour dateFacturation et pour dateCommande

▶ absence de **synonyme** : noms différents pour une même valeur.

✓ Document Facture :

N° de facture	12381	Nom de magasin	Alpha-Lille	
N° Magasin	12	Nom Client	Martin	
N° client	12345	Adresse Client	5 rue Merise Bat 4 Entree 7 59000 Lille	
N° produit	Libellé produit	Quantité	Prix unitaire HT	Prix HT
452	Merise pour les nuls	130	100.00	13000.00
654	Plus loin avec Merise	1	200.00	200.00
1093	Vidéo K7 Merise	10	150.00	1500.00
87	Merise air bag	1	6000.00	6000.00
Total HT	13200	Montant TVA	726.00	
Total HT	1500	Montant TVA	330.00	
Total HT	6000	Montant TVA	1236.00	
Montant TTC	22992.00			

✓ Document Etat des Stocks :

N° magasin	12	
N° produit	Désignation produit	Quantité en Stock
87	Merise air bag	128
452	Merise pour les nuls	4387

Dictionnaire global des informations

Nom	Nature	Type	Document	Observations
N° facture	CM	N	Facture	
N° magasin		N	Facture	
Nom magasin		A	Facture	
Nom client		A	Facture	
N° client	CM	N	Facture	
Adresse 1		A	Facture	Norme postale
Adresse 2		A	Facture	Norme postale
Adresse 3		A	Facture	Norme postale
Code postal		N	Facture	Norme postale
Ville		A	Facture	Norme Postale
N° produit	CM	N	Facture	
libellé produit		A	Facture	
Quantité		N	Facture	
Prix unitaire		N	Facture	
Prix HT	C	N	Facture	Quantité × prix Unitaire
Total HT	C	N	Facture	pour chaque Taux TVA id : \sum Prix HT
Montant TVA	C	N	Facture	Total HT × Taux TVA
Montant TTC	C	N	Facture	\sum (Montant TVA + Total HT)
N° produit	CM	N	Stocks	
Désignation produit		A	Stocks	
Quantité en stocks	CM	N	Stocks	\sum (sommeEntrées – sommeSorties)

Modification du dictionnaire

- ✓ En fonction des polysèmes, synonymes :
 - ▶ Elimination de (n°magasin, n°produit)
 - ▶ Synonyme : DésignationProduit et libelléProduit
- ✓ En supprimant les données calculées
- ✓ En ajoutant des informations détectées par les règles de calcul :
 - ▶ Introduction de tauxTVA

Dictionnaire épuré des informations

Nom	Nature	Type	Observations
N° facture	CM	N	
N° magasin		N	
Nom magasin		A	
Nom client		A	
N° client	CM	N	
Adresse 1		A	Norme postale
Adresse 2		A	Norme postale
Adresse 3		A	Norme postale
Code postal		N	Norme postale
Ville		A	Norme Postale
N° produit	CM	N	
libellé produit		A	
Quantité		N	
Prix unitaire		N	
Taux TVA		N	
Quantité en stocks	CM	N	$\sum(\text{sommeEntrées} - \text{sommeSorties})$
Entrée stock		N	

Remarque : AMC-Designor permet d'établir ce dictionnaire

Passage au schéma conceptuel \implies vérification automatique que toutes les données sont traitées.

Phase II : Structuration logique du dictionnaire

✓ Notion de dépendance fonctionnelle :

Définition 1. *Une information B dépend fonctionnellement d'une information A ssi à une seule valeur de A , il n'est possible d'associer qu'une et qu'une seule valeur de B .*

✓ Astuce : trouver les identifiants (clés)

⇒ Dépendances fonctionnelles directes

Dépendances fonctionnelles directes

Nom	dépend de
N° facture	
N° magasin	N° facture
Nom magasin	N° magasin
Nom client	N° client
N° client	N° facture
Adresse 1	N° client
Adresse 2	N° client
Adresse 3	N° client
Code postal	N° client
Ville	N° client
N° produit	
libellé produit	N° produit
Quantité	
Prix unitaire	N° produit
Code TVA	N° produit
Taux TVA	Code TVA
Quantité en stocks	
Entrée stock	

Remarque : introduction de l'information codeTVA : chaque produit est soumis à un taux de TVA. Plus facile pour les mises à jour éventuelles du taux de TVA.

Phase II : Structuration logique du dictionnaire (suite)

- ✓ Notion de dépendances fonctionnelles composées :
- ✓ Deux possibilités :
 - ▶ L'information dépend fonctionnellement de plusieurs autres informations : la dépendance fonctionnelle est composée.
 - ▶ L'information ne possède pas de dépendance, alors l'information dépend fonctionnellement d'elle-même.

⇒ Dépendances fonctionnelles directes et composées

Dépendances fonctionnelles directes et composées

Nom	dépend de
N° facture	N° facture
N° magasin	N° facture
Nom magasin	N° magasin
Nom client	N° client
N° client	N° facture
Adresse 1	N° client
Adresse 2	N° client
Adresse 3	N° client
Code postal	N° client
Ville	N° client
N° produit	N° produit
libellé produit	N° produit
Quantité	N° produit, N° facture
Prix unitaire	N° produit
Code TVA	N° produit
Taux TVA	Code TVA
Quantité en stocks	N° produit, N° magasin
Entrée stock	N° produit, N° magasin

Les Groupes Logiques d'Information : G.L.I.

Définition 2. *Un groupe logique d'information (GLI) est un ensemble d'informations possédant un même identifiant qu'il soit élémentaire ou composé*

- ✓ Une information A est un identifiant pour une information B ssi B dépend fonctionnellement de A et aucune information différente de B ne dépend de B.
- ✓ Plusieurs informations (I_1, I_2, \dots, I_n) constituent un identifiant composé pour une information B ssi B dépend fonctionnellement de chacune des informations I_i et aucune information différente de B ne dépend de B

Les Groupes Logiques d'information

✓ Quelques exemples :

- ▶ $n^{\circ}\text{Magasin}$ est un identifiant pour nomMagasin
- ▶ $n^{\circ}\text{Facture}$ n'est pas un identifiant pour $n^{\circ}\text{Magasin}$:
 - $n^{\circ}\text{Magasin}$ dépend bien de $n^{\circ}\text{Facture}$ mais nomMagasin dépend de $n^{\circ}\text{Magasin}$

⇒ Etablir le graphe de dépendances

Les Groupes Logiques d'Information

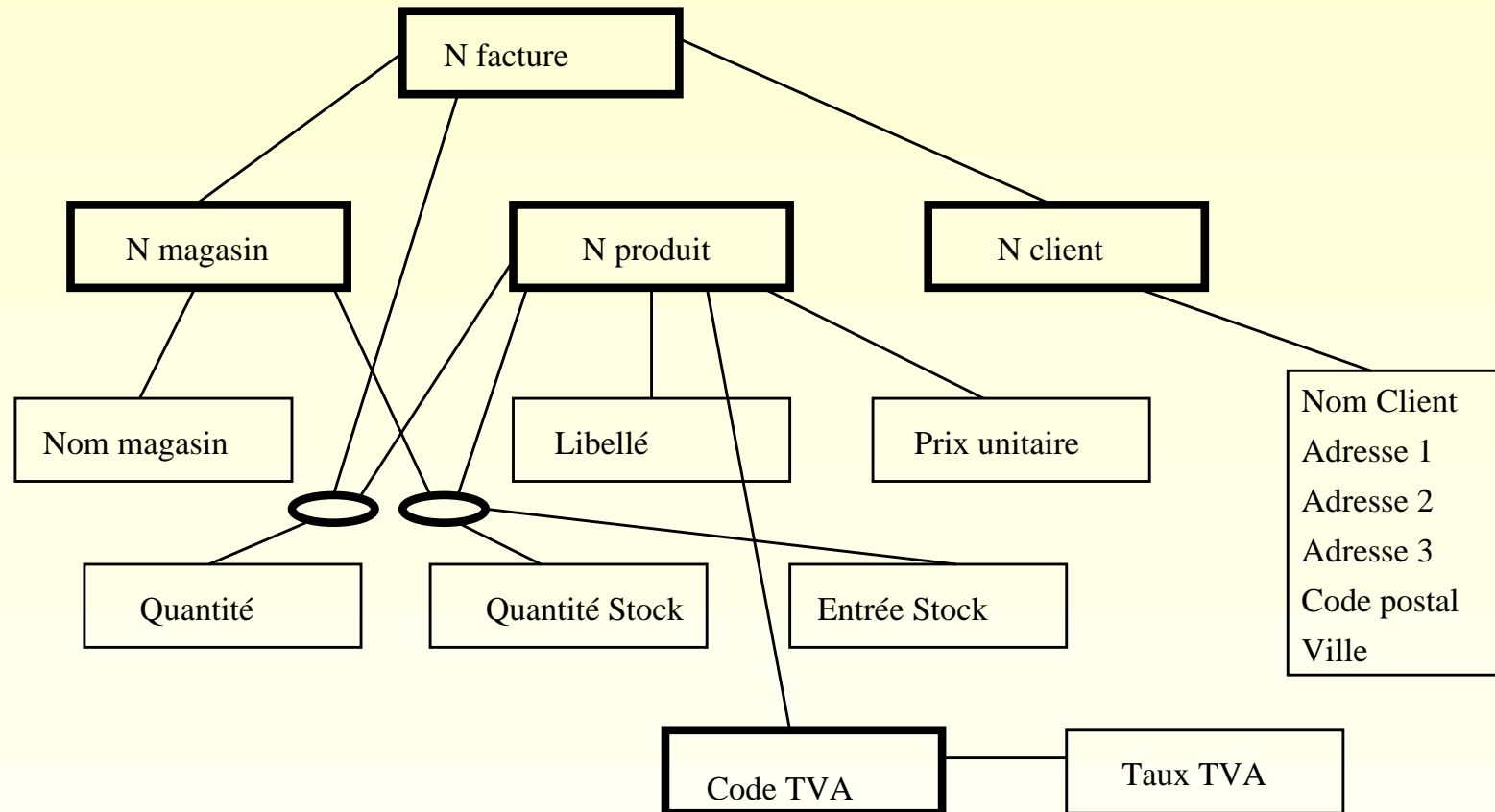


Tableau des GLI

- ✓ Une autre représentation peut être utilisée sous forme de tableau des GLI. Les symboles utilisés sont :
 - ▶ I : Identifiant d'un GLI
 - ▶ IC : Identifiant composé d'un GLI
 - ▶ x : information appartenant à un GLI
 - ▶ A : association entre GLI
- ✓ Plus simple à décrire avec un outil informatique.

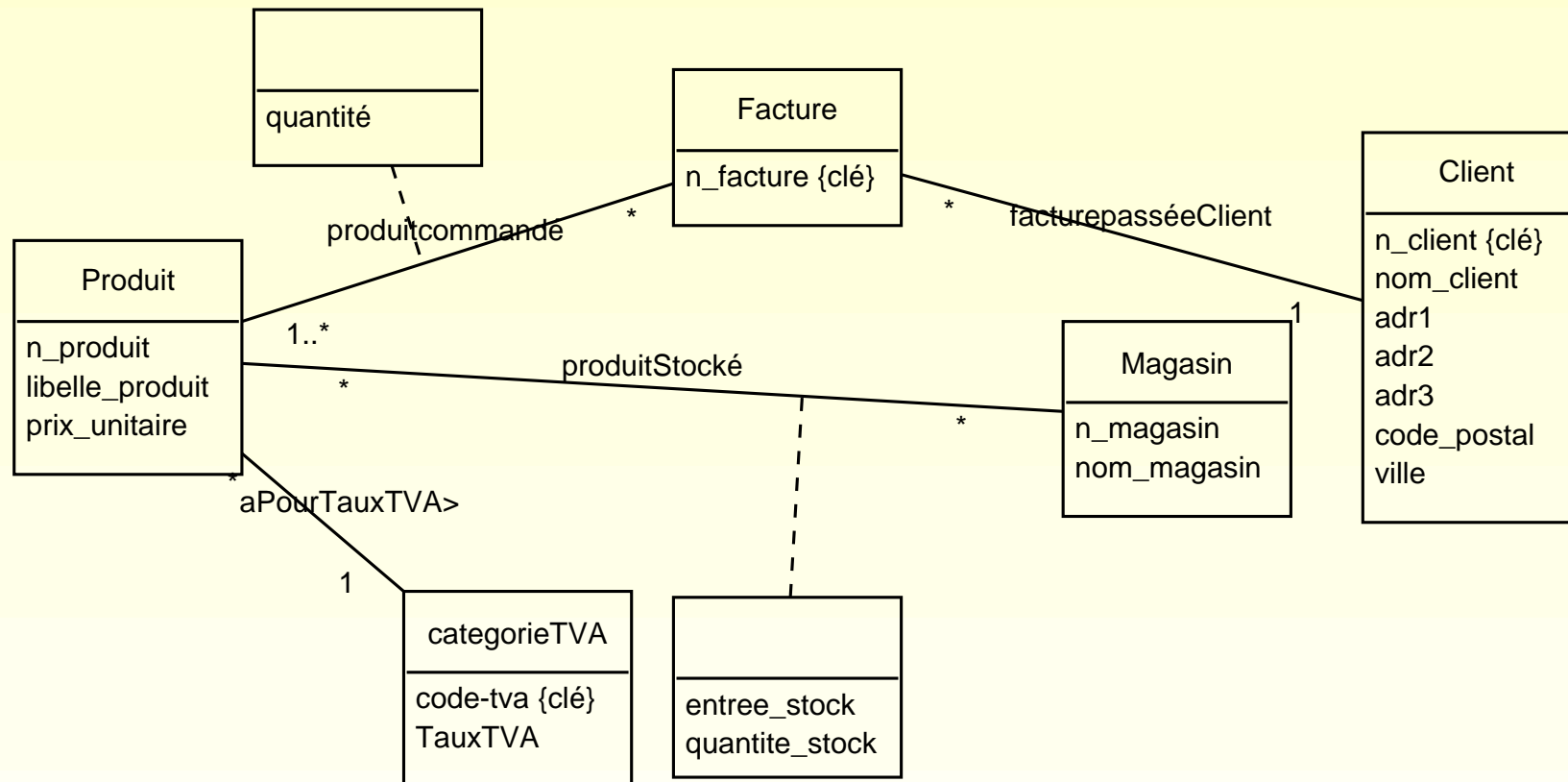
Tableau des GLI

Nom	Facture	Magasin	Client	Produit	TVA	Prod-Mag	Prod-Fact
N° facture	I						IC
N° magasin	A	I				IC	
Nom magasin		x					
Nom client			x				
N° client	A		I				
Adresse 1			x				
Adresse 2			x				
Adresse 3			x				
Code postal			x				
Ville			x				
N° produit				I		IC	IC
libellé produit				x			
Quantité							x
Prix unitaire				x			
Code TVA				A	I		
Taux TVA					x		
Quantité en stocks						x	
Entrée stock						x	

Apport des GLI

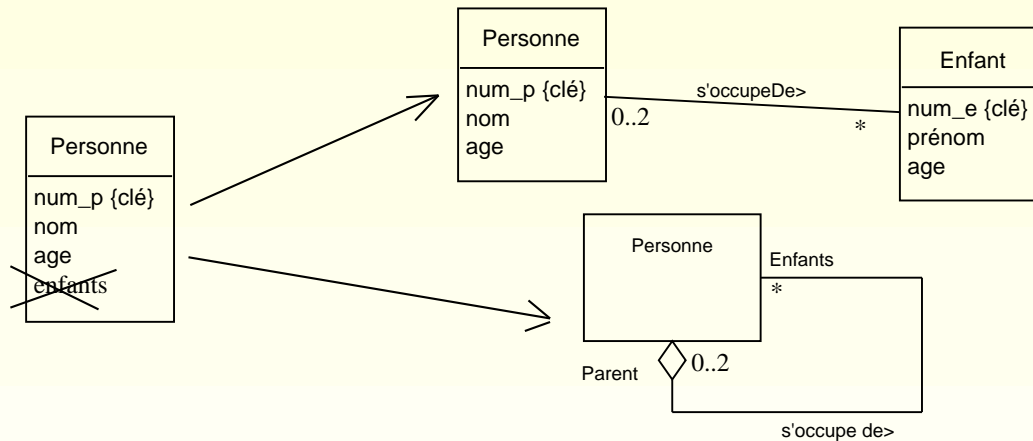
- ✓ Faciliter la construction d'un schéma conceptuel (basé sur un modèle de type Entité-Association)
- ✓ Vérifier la cohérence du modèle :
 - ▶ Il existe un et un seul x ou l sur une ligne donnée
 - ▶ Un identifiant composé (IC) possède bien un identifiant (I) dans un autre GLI
 - ▶ Une information d'association (A) possède bien un identifiant (I) dans un autre GLI
 - ▶ Pour toute colonne, il doit exister un identifiant unique (I) ou plusieurs identifiants IC formant un identifiant composé.

Le schéma conceptuel UML de l'exemple



Règles de validation d'un schéma conceptuel (1/3)

- ✓ **non redondance** : une propriété figure dans une seule entité ou association.
 - ✓ **première forme normale** : chaque propriété d'une entité ou d'une association possède au plus une valeur à un instant t.
- Ex : enfants d'une personne (propriété multivaluée)

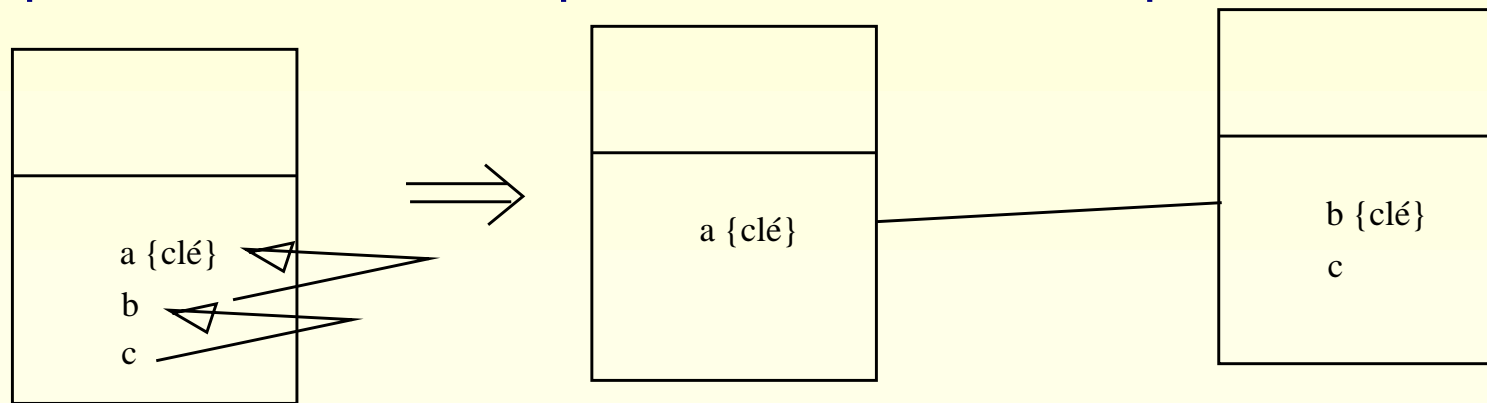


Règles de validation d'un schéma conceptuel (2/3)

- ✓ **deuxième forme normale** : chaque propriété d'une entité dépend de l'identifiant de l'entité et chaque propriété d'une association dépend de tous les identifiants des entités qui composent l'association.
- ✓ conforme à la conception des G.L.I. (Groupe Logique d'Informations)

Règles de validation d'un schéma conceptuel (3/3)

- ✓ **troisième forme normale** : chaque propriété d'une entité dépend d'un identifiant (entité) ou de plusieurs identifiants (association) et non d'une propriété elle-même dépendante d'un ou de plusieurs identifiants



- ✓ Attention : les formes normales ont un sens pour le modèle relationnel, pas pour d'autres modèles (ex : modèle objet-relationnel)