

Conception d'un système
d'information:
Méthode de conception Merise

Hala Skaf-Molli

Skaf@loria.fr

<http://www.loria.fr/~skaf>

B 032

Sources

- Cours de Claudine Toffolon (Université du Littoral)
- Frédéric Julliard (Université de Bretagne Sud, IUP Vannes)
- Cours Gilles Simon (Université Henri Poincaré, Nancy1)
- Merise et UML pour la modélisation des systèmes d'information-Un guide complet avec études d - Gabay, Joseph-4e éd-2001-02
- Bases de données et systèmes d'information, Nacer Boudjlida, Dunod, 1999.
- Merise, méthode de conception. A. Collongues, J. Hugues, B. Laroche, Dunod, 1986.
-

Historique

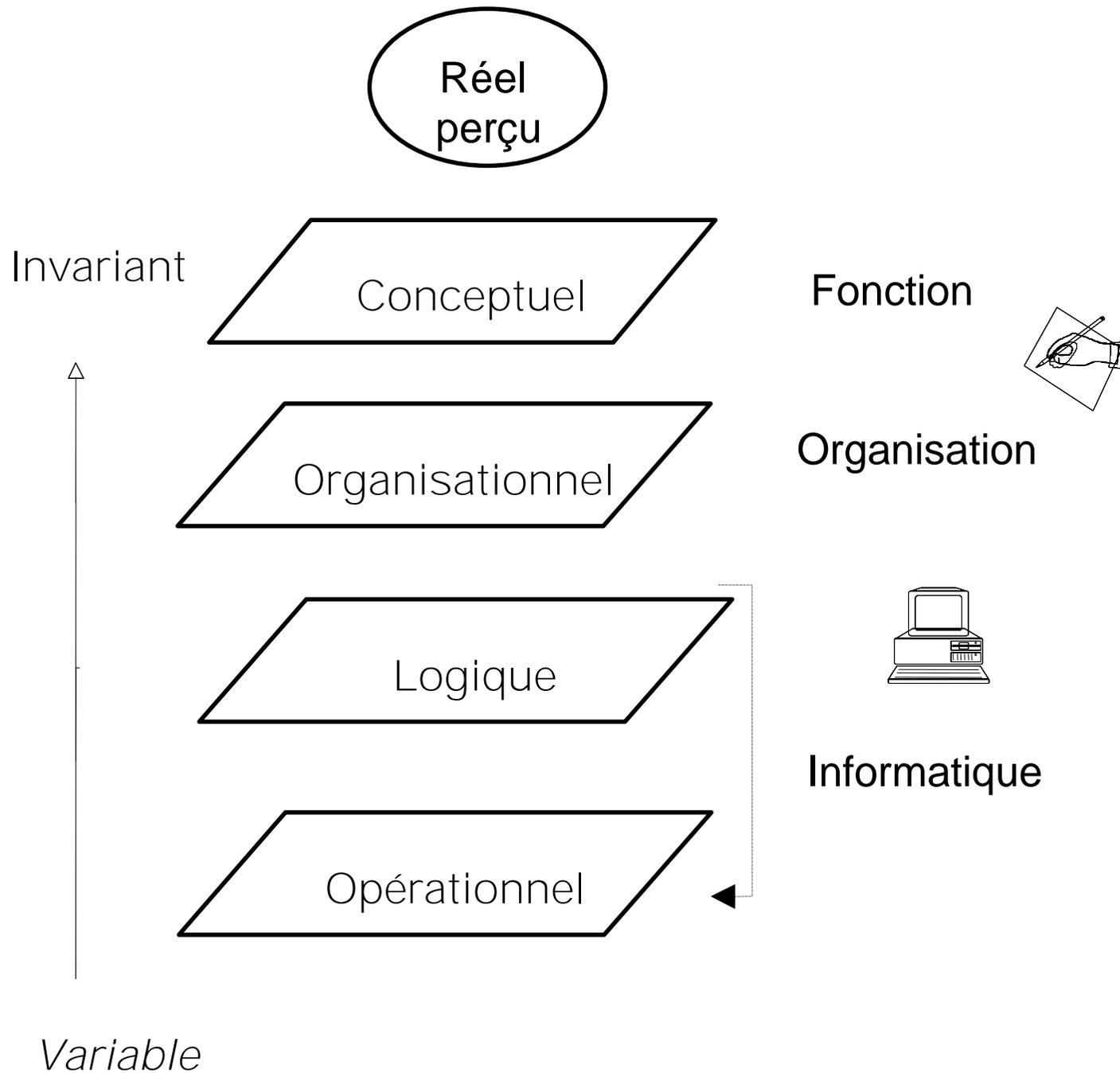
- Approche ancienne : 1978
- Très répandue en France
- Origine française : développée par :
 - CTI (Centre Technique d'Informatique)
 - CETE(Centre d'Etudes Techniques de l'Equipement)
- Remise à jour : Merise 2
 - à la mode «Objet» : concepts de classes, héritage...

Caractéristiques

- Vision globale de l'entreprise
- Séparation des données et des traitements
 - Traitements:
 - Étude des évènements
 - Indépendances entre les domaines
 - Données
 - Étude du vocabulaire de l'organisation
 - Intégration des domaines: Vue globale
- Approche par niveaux :
 - Quatre niveaux de description ou niveaux d'abstraction

Approche par Niveaux

- NIVEAU CONCEPTUEL: Ce qu'il faut faire
 - QUOI ?
- NIVEAU ORGANISATIONNEL: La manière de faire
 - QUI ?, QUAND ?, COMBIEN ?, OU ?
- NIVEAU LOGIQUE: Choix des moyens et ressources
 - AVEC QUOI ? QUELS OUTILS ?
- NIVEAU PHYSIQUE: Les moyens de le faire
 - COMMENT ?



Niveau Conceptuel

- Exprime les choix fondamentaux de gestion, les objectifs de l'organisation
- Décrit les invariants de l'organisation:
 - le métier de l'organisation
- Indépendamment
 - des aspects organisationnels
 - des aspects techniques de mise en oeuvre
- du point de vue:
 - des traitements: objectif, résultat, règles de gestion, enchaînement
 - des données: signification, structure, liens
- C'EST LA DESCRIPTION LA PLUS STABLE DU SYSTEME

Niveau Organisationnel

- La répartition géographique et fonctionnelle des sites de travail (du point de vue des données et des traitements)
- le mode de fonctionnement: temps réel ou temps différé
- la répartition du travail homme/machine (degré et type d'automatisation)
- les postes de travail et leur affectation, la volumétrie des données, la sécurité des données
- Indépendamment des moyens de traitement et de stockage de données actuels ou futurs
- C'est la description des postes de travail de l'entreprise et des informations qu'elle traite.

Niveau Logique

- Exprime la forme que doit prendre l'outil informatique pour être adapté à l'utilisateur, à son poste de travail
- Indépendamment de l'informatique spécifique, des langages de programmation ou de gestion des données
- Décrit
 - le schéma de la base de données (relationnel, hiérarchique ou réseau) ie- les caractéristiques du mode de gestion des données
 - la répartition des D sur les différentes unités de stockage
 - les volumes par unité de stockage
 - l'optimisation des coûts induits par le mode de gestion

Niveau Physique

- Traduit les choix techniques et la prise en compte de leurs spécificités
- Répond aux besoins des utilisateurs sur les aspects logiciels et matériels.
- Définit complètement:
 - les fichiers, les programmes
 - l'implantation physique des données et des traitements,
 - les ressources à utiliser,
 - les modalités de fonctionnement
- C'EST LA DESCRIPTION DES MOYENS MIS EN OEUVRE POUR GERER LES DONNEES ET EFFECTUER LES TRAITEMENTS.

Approche par Niveaux

- Les niveaux conceptuel et organisationnel représentent toute l'organisation
- Les niveaux logique et physique ne prennent en compte que la solution informatique

Approche par Niveaux

- A chaque niveau correspond un modèle
- **MODELE = SCHEMA + DESCRIPTIF**
- **SCHEMA NORMALISE**
 - Synthèse
 - Communication
- **DESCRIPTION TEXTUELLE**
 - Définitions
 - Commentaires
 - Quantifications
 - Contraintes

La modélisation

- Un modèle doit posséder au moins trois qualités:
 - La fidélité: la représentation doit être effectuée sans déformation de la réalité
 - La cohérence: la représentation ne doit comporter de contradiction explicite ou implicite
 - La complétude: la représentation doit décrire tous les phénomènes pertinents par rapport aux objectifs du modélisation, ce qui n'est pas synonyme d'exhaustivité systématique

Les Modèles au niveau conceptuel

- Le Modèle Conceptuel des Données : M.C.D.
 - Description des données et des relations en termes:
 - ENTITE ou INDIVIDU
 - RELATION ou ASSOCIATION
 - PROPRIETES ou ATTRIBUT

- Le modèle Conceptuel des Traitements : M.C.T.
 - Description de la partie dynamique du S.I. en termes
 - PROCESSUS
 - OPERATION comprenant les concepts d'EVENEMENT /RESULTAT et de SYNCHRONISATION

Les Modèles au niveau Organisationnel/Logique

- Le Modèle logique de données: M.L.D.
 - *Le modèle CODASYL si une orientation base de données réseau est choisie*
 - *Le modèle RELATIONNEL si une orientation base de données relationnelle est choisie*
 - *Le modèle HIERARCHIQUE*
- Le Modèle Organisationnel des Traitements: M.O.T
 - permet de représenter par procédure les phases et les tâches effectuées par chaque poste de travail

Les Modèles au niveau Physique ou Opérationnel

- Le Modèle Physique des Données : M.P.D
 - spécifie les organisations physiques de données
- Le Modèle Physique des Traitements: M.P.T
 - décrit les traitements réalisés pour chaque transaction (temps réel) ou chaque unité de traitement (temps différé)

Les Concepts de MERISE

Niveau de description	Concepts Manipulés		
	<i>Données</i>		<i>Traitements</i>
<i>Conceptuel</i>	<ul style="list-style-type: none"> • Entité/ Individu • Association <ul style="list-style-type: none"> • <i>Propriétés</i> • <i>Contrainte</i> <p style="text-align: center;">M . C . D</p>		<p>Processus</p> <p>Opération</p> <ul style="list-style-type: none"> • <i>Évènement/ Résultat</i> • <i>Synchronisation</i> • <i>Règles de gestion</i> <p style="text-align: center;">M . C . T</p>
<i>Organisationnel / Logique</i>	<ul style="list-style-type: none"> • Modèle relationnel <i>Tables, Attributs</i> • Modèle Codasyl <i>Record, Champs, Set</i> • Modèle hiérarchique <p style="text-align: center;">M . L . D</p>		<p>Procédure</p> <p>Phase</p> <p>Tâche</p> <p style="text-align: center;">M . O . T</p>
<i>Physique/ Opérationnel</i>	<ul style="list-style-type: none"> • Tables, Tuple, Attributs • Langage SQL <p style="text-align: center;">M . P . D</p>	<ul style="list-style-type: none"> • Record, Article, Champs, Set • Langages spécifiques SGBD <p style="text-align: center;">M . P . D</p>	<p>Application</p> <p>Unité de traitement</p> <ul style="list-style-type: none"> • Temps réel : <i>Transaction</i> • Temps différé : <i>Programme Batch</i> <p style="text-align: center;">M . P . T .</p>

L a D o u b l e A p p r o c h e :

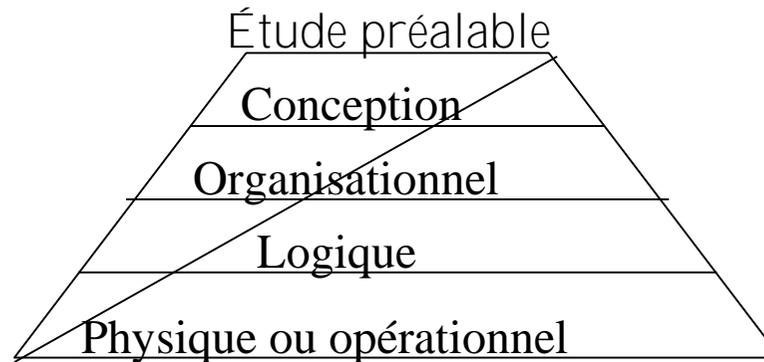
N i v e a u x e t M o d è l e s

N i v e a u x	M o d è l e s	
	<i>D O N N E E S</i>	<i>T R A I T E M E N T S</i>
Conceptuel	M o d è l e C o n c e p t u e l d e s A c t i v i t é s (M C A)	
	M C D	M C T
	<div style="border: 2px solid black; padding: 2px; display: inline-block;">V a l i d a t i o n</div> 	
Organisationnel	M O D	M O T
	<div style="border: 2px solid black; padding: 2px; display: inline-block;">V a l i d a t i o n</div> 	
Logique	M L D	M L T
	<div style="border: 2px solid black; padding: 2px; display: inline-block;">V a l i d a t i o n</div> 	
Physique/ Opérationnel	M P D	M P T
	<div style="border: 2px solid black; padding: 2px; display: inline-block;">V a l i d a t i o n</div> 	

T r o i s v o i e s d e v a l i d a t i o n

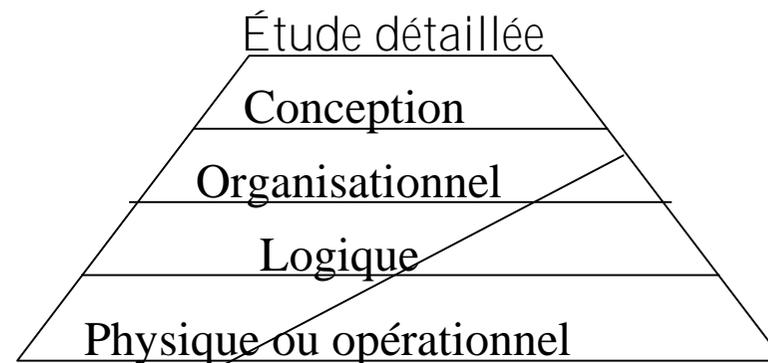
- *Par les règles de gestion*
- *Mise En Cohérence des modèles*
- *Description des Evènements/Résultats*

Étapes de développement d'un SI

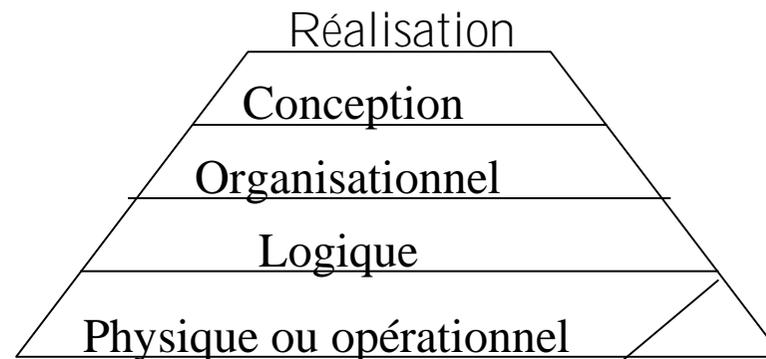


L'étude préalable ne traite pas tous les cas particuliers...

Niveau couvert
Par la description du
Système d'information



Une petite partie des spécifications détaillées est traitée dans la phase de réalisation



Un logiciel ne peut être testé à 100% à la fin de la réalisation..

Modèle conceptuel des traitements

- SCT (ou MCT - Modèle ...) est une abstraction des activités du système d'information et de leurs contraintes
- Inspiré des réseaux de pétri
- Processus de conception :
 - Modèle conceptuel de communication
 - Identification des acteurs et des flux d'informations
 - Ordonnancement des flux
 - Elaboration du SCT

Modèle conceptuel de communication (MCC)

1. Définir l'organisation

Objectif, activités, produits, clients, décision, finances...

2. Établir le modèle de contexte

Donner le cadre de l'étude

Vue synthétique du problème

3. Établir le modèle conceptuel de flux

fixer la portée et les limites du futur système ou pour le décomposer en sous-systèmes

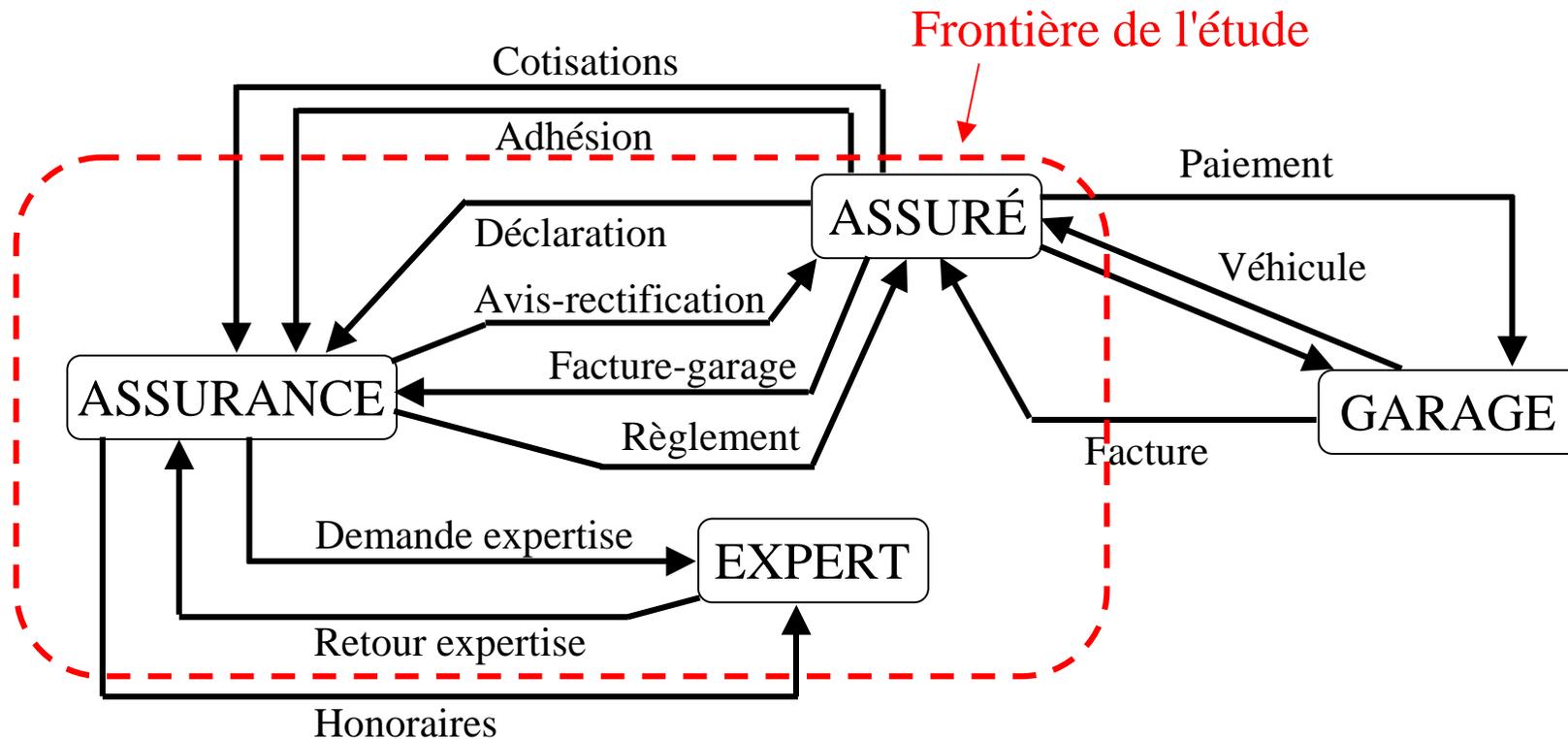
Se présente sous la forme d'un graphe dont les nœuds sont des acteurs identifiés du SI et les arcs montrent les types d'information circulant entre les acteurs

4. Diagramme de dépendance des documents

Exemple

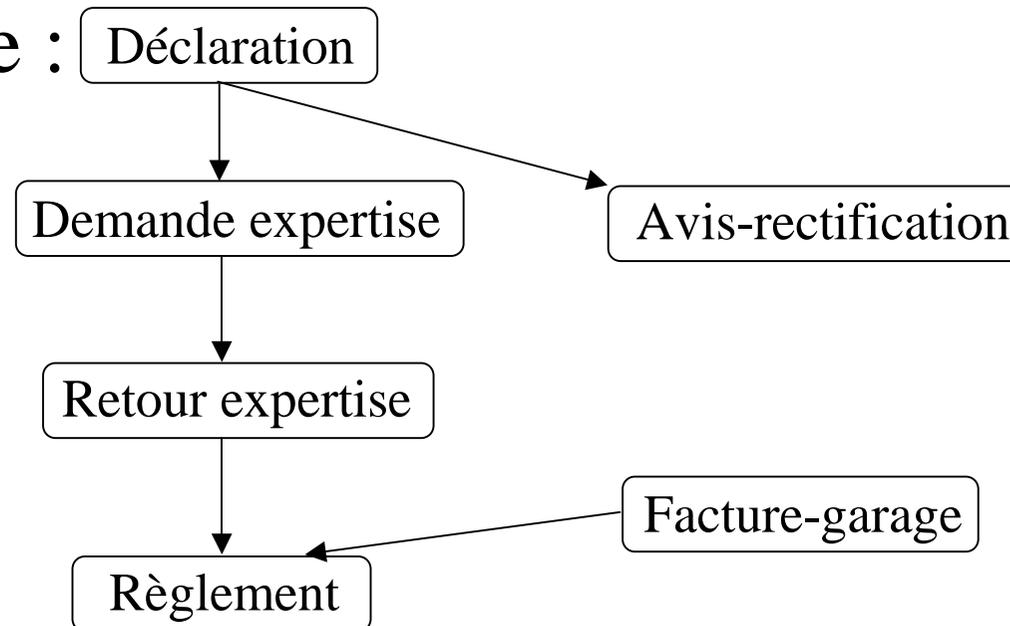
- Traitement d'un sinistre automobile par une compagnie d'assurance:
 - Toute déclaration incorrecte n'est pas enregistrée. Elle entraîne l'émission d'un avis au sinistré qui devra faire une nouvelle déclaration
 - Un expert donne son avis....
 - Le règlement du sinistre ne se fait qu'après réception de la facture du garage ayant effectué les réparations
 - En fin d'année archiver tous les dossiers traités

MCF de la gestion d'accident par une assurance..



Le graphe de précédence « MOF »

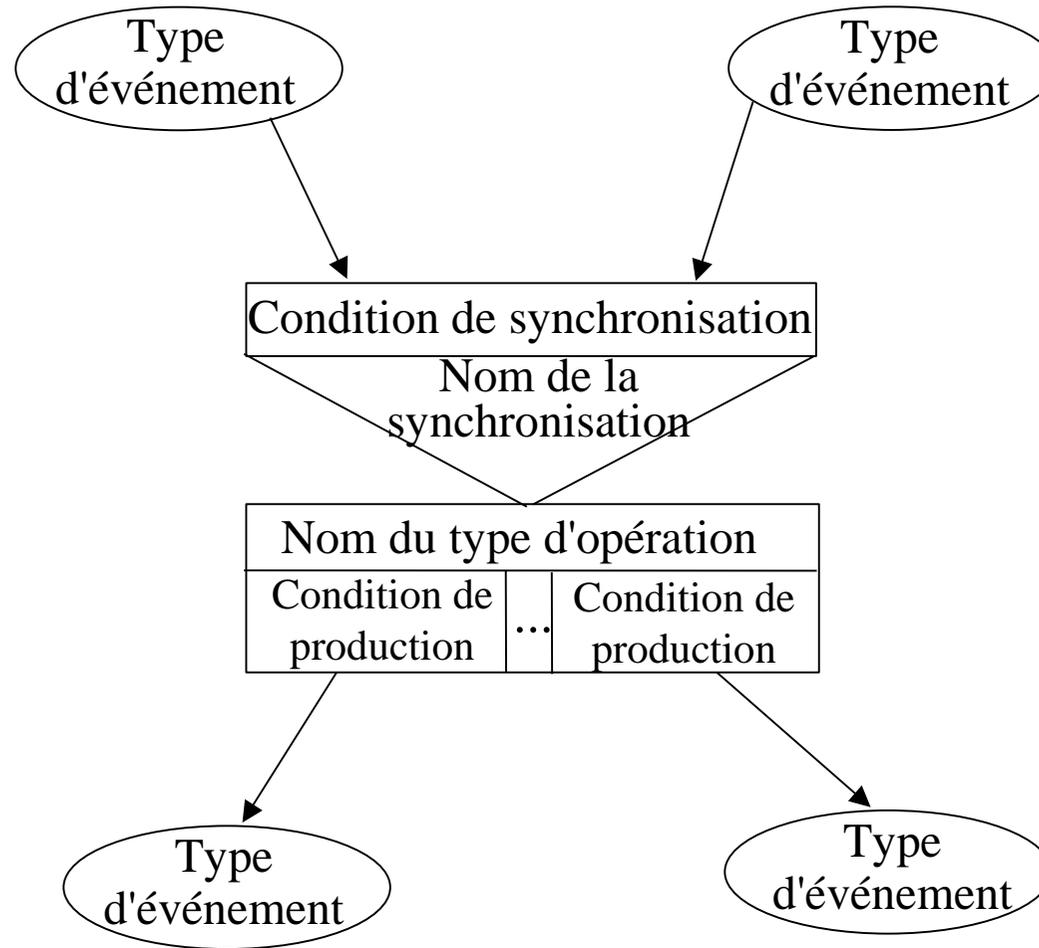
- Montre les dépendances *temporelles* entre les types d'informations
- Exemple :



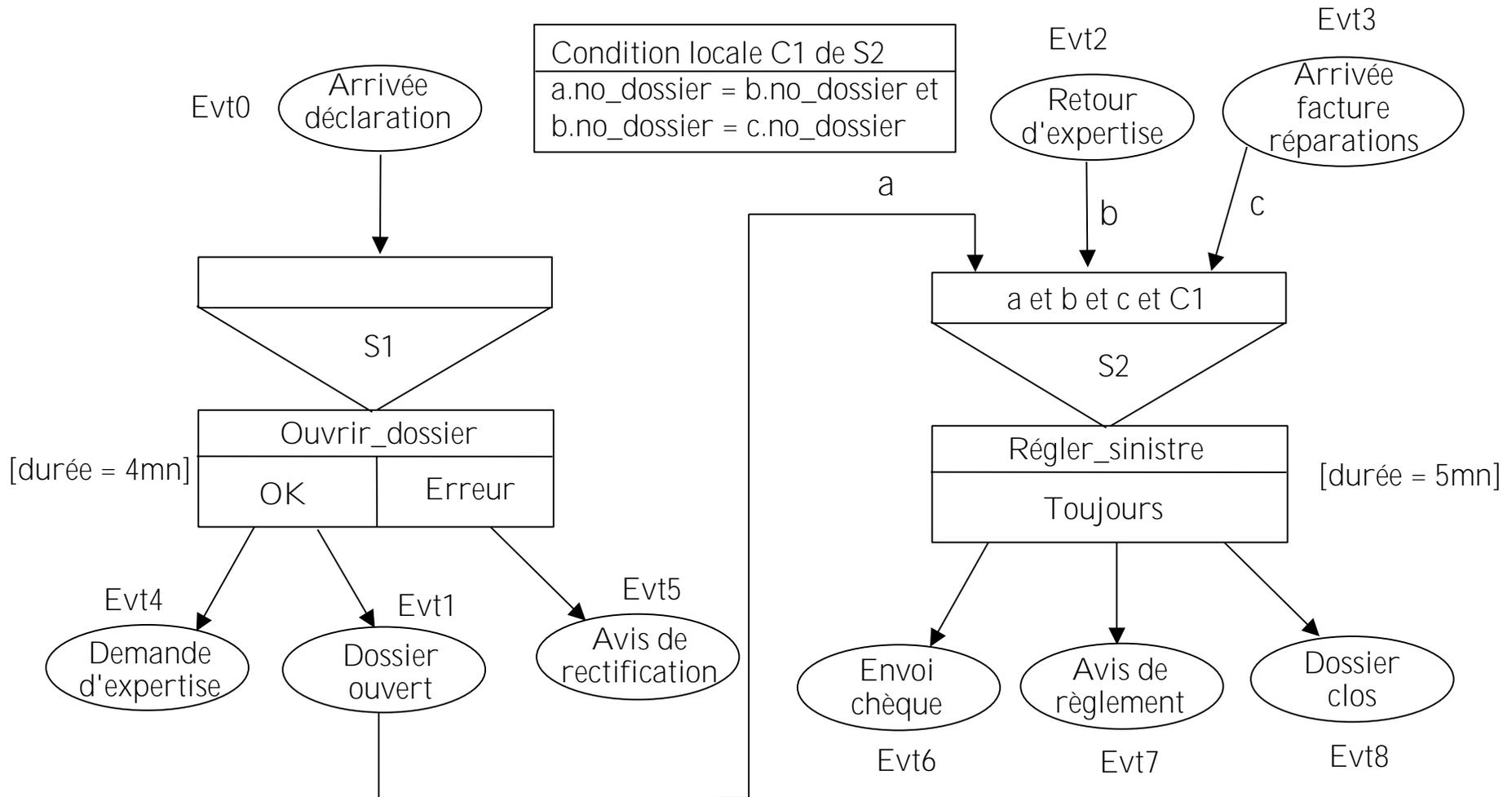
Élaboration du schéma conceptuel des traitements

- En se basant sur le graphe ordonné des flux, on introduit les traitements concernant un ou plusieurs flux
- On décrit le rôle des traitements et les informations nécessaires en entrée et produites en résultat

Représentation graphique du modèle des traitements



Exemple de SCT



Type d'événement

- Description lexicale :
 - *nom et message*
 - *identifiant* des occurrences
 - *fréquence* d'apparition au cours d'une période donnée
 - *capacité* (nb max d'occurrences que le SI peut prendre en compte au cours d'une période)
 - liste des *synchronisations* auxquelles il participe et des opérations qu'il peut déclencher

Exemple de type d'événement

- Événement *Arrivée déclaration* (Evt0)
 - *message* : informations figurant sur la déclaration
 - *identifiant* : le couple (no de l'assuré, date d'arrivée de la déclaration)
 - *fréquence* : 50 par jour
 - *capacité* : 55 par jour
 - participe à la *synchronisation* S1 du type d'opération Ouvrir_dossier
- La description du message peut être formalisée :
<Nom_assuré : chaîne, Prénom : chaîne, No_police : chaîne, Date_accident : date ...>

Type d'opération

- Description lexicale :
 - nom et rôle
 - durée
 - type(s) d'événements qui conditionnent son déclenchement (entrées)
 - type(s) d'événements produits (sorties)
 - si la production des événements est conditionnelle, expliciter la condition de production de chaque événement
 - action réalisée

Exemple de type d'opération

- Opération *Ouvrir_dossier*
 - *Rôle* : Vérifie une déclaration et initialise l'expertise
 - *Durée* : 10 minutes
 - *Événements en entrée* : Evt0
 - *Événements en sortie* : (Evt4 et Evt1) ou Evt5
 - *Action* :
 - si déclaration_OK
 - alors Ouvrir un dossier de sinistre (Evt1)
 - Faire une demande d'expertise de ce dossier (Evt4)
 - sinon Renvoyer la déclaration à l'assuré (Evt5)
 - fsi

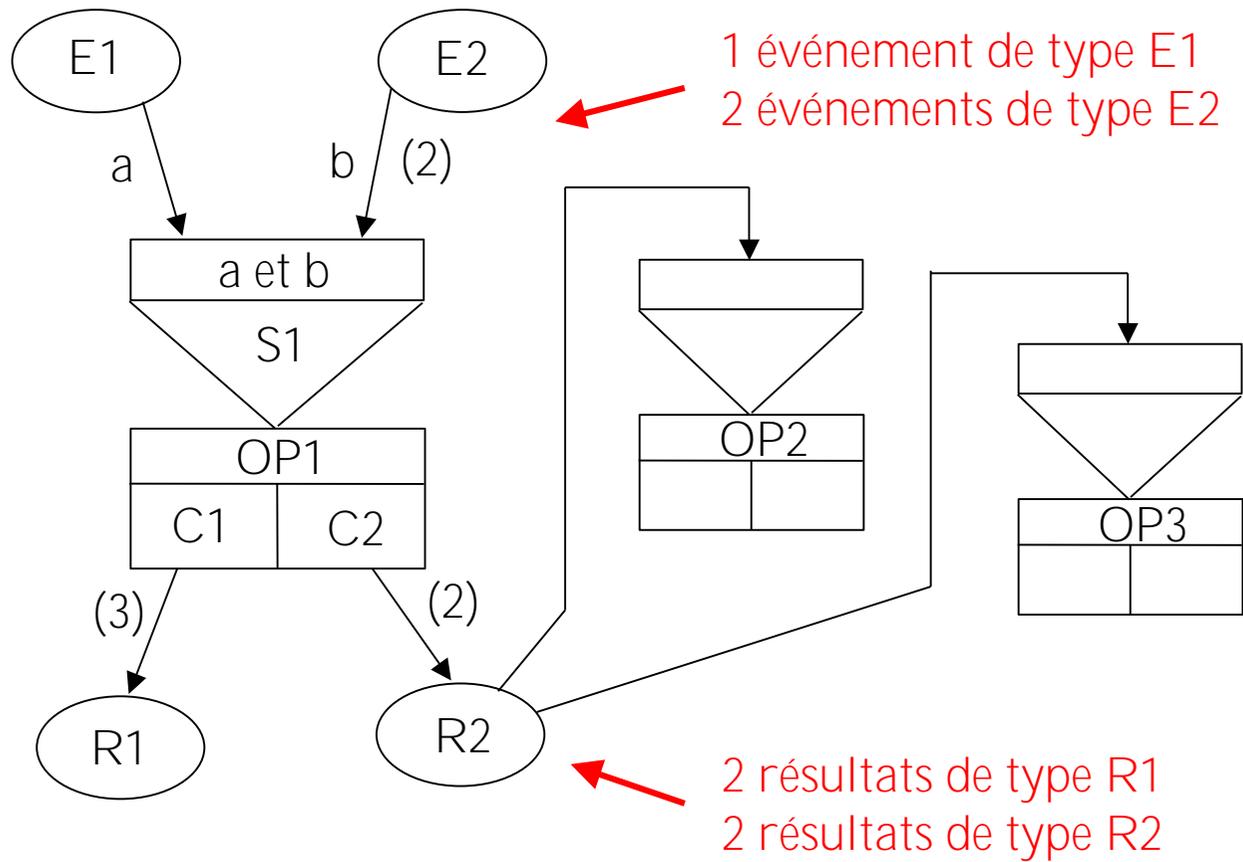
Type de synchronisation

- Description lexicale :
 - *nom*
 - *liste des types d'événements qui participent à la synchronisation*
 - éventuellement, *condition de synchronisation* portant sur les types d'événements
 - *condition locale* : précise, en présence de plusieurs occurrences d'un type d'événements, laquelle choisir
 - *délai de synchronisation* : temps max séparant le moment où la synchronisation est activable et celui où elle est activée
 - *durée limite* : temps max d'attente entre l'arrivée du premier événement et celle du dernier

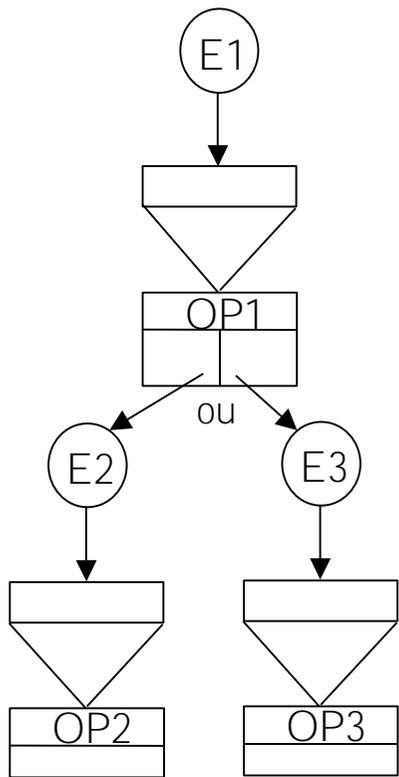
Exemple de type de synchronisation

- Synchronisation S2
 - Condition : $\text{Evt1 (a)} \wedge \text{Evt2 (b)} \wedge \text{Evt3 (c)} \wedge \text{C1}$
 - Condition locale C1 :
 - $\text{a.no_dossier} = \text{b.no_dossier} \wedge$
 $\text{b.no_dossier} = \text{c.no_dossier}$
 - et "premier arrivé premier servi"
 - Délai de synchronisation : Trois jours
 - Durée limite : douze mois

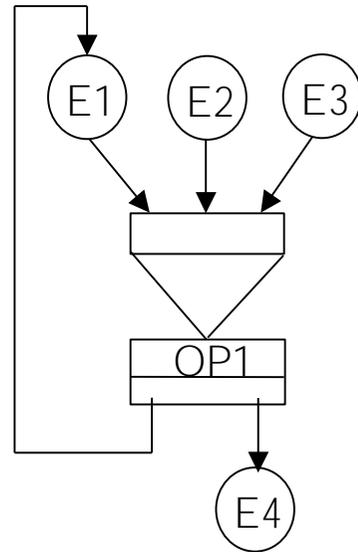
Nombre d'occurrences



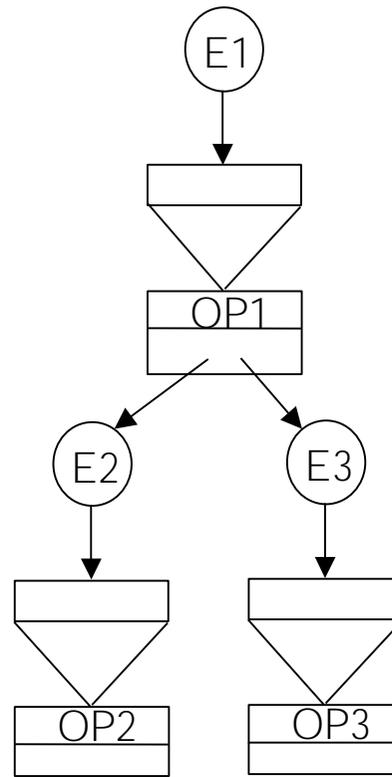
Structures de base d'un SCT



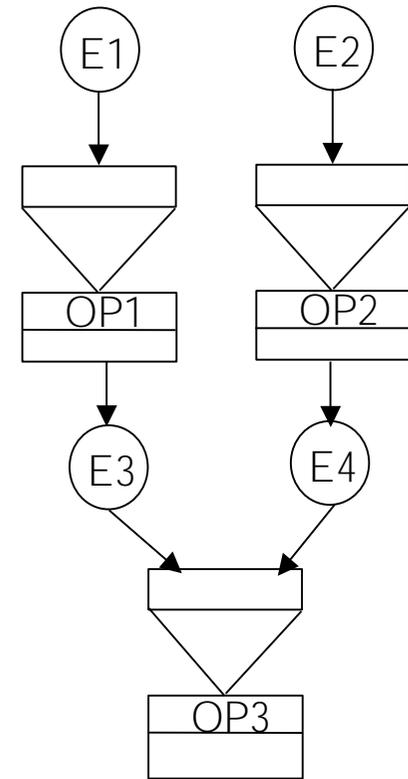
Alternative



Itération



*Parallèle
divergente*



*Parallèle
convergente*

Démarche pratique pour la modélisation conceptuelle

- Élaborer et ordonner un diagramme de flux
- Faire une première ébauche de SCD
- Faire une première ébauche de SCT
- Pour chaque opération du SCT, analyser ses effets sur le SCD
- Modifier ou compléter le SCD
- Modifier ou compléter le SCT
- Itérer sur les trois étapes précédentes

Le schéma logique des traitements

- Aboutit à une architecture de déploiement du système, obtenue par raffinement des opérations conceptuelles
- Les étapes de la construction du SLT :
 - décomposer les opérations du SCT en sous-opérations appelées *procédures* ou *fonctions*
 - affecter et localiser chaque procédure
 - détailler l'analyse de chaque procédure
 - définir l'enchaînement des procédures
 - estimer le coût de mise en place de la base
 - essayer de réduire ce coût

Décomposition des opérations

Exemple : l'opération *Ouvrir_dossier* peut être décomposée en les procédures suivantes :

- vérifier la déclaration (assuré connu, circonstances bien décrites ...)
- l'ignorer ou lui affecter un numéro de dossier
- enregistrer les informations nécessaires dans la base
- désigner un expert pour le nouveau dossier
- transmettre le dossier à l'expert

Identification des procédures

Pour chaque procédure sont fournis :

- un nom
- un mode de réalisation (manuelle, automatisée totalement ou partiellement, interactive, différée ...)
- une localisation (où?)
- une affectation (qui?)
- une fréquence d'activation

Exemple

Nom	No	automa- tisable ?	Mode	Localisation/ affectation
Vérifier_déclaration	P1	non	manuel	Hôtesse
Attribuer_no_dossier	P2	oui	conversationnel	Hôtesse
Enregistrer_dossier	P3	non	conversationnel	Hôtesse
Désigner_expert	P4	non	conversationnel	Chef de service
Transmettre_dossier	P5	non	manuel	Secrétariat du chef de service

Analyse détaillée des procédures

Décrire :

- les événements ou données nécessaires au déclenchement de la procédure et les résultats qu'elle produit
- les traitements effectués et les actions réalisées sur la base :
algorithme + algèbre relationnel à partir du SLD
- les supports des données et des résultats (formulaire papier, écrans de dialogue etc.)

Exemple

SLD Assuré(no_ass, nom_ass, adr_ass, tel_ass, no_agence)

Fonction *vérifier_déclaration*

Données

d : déclaration

Début

Si $\sigma_{\text{no_ass}} = \text{d.no_police}$ (Assuré) = { }

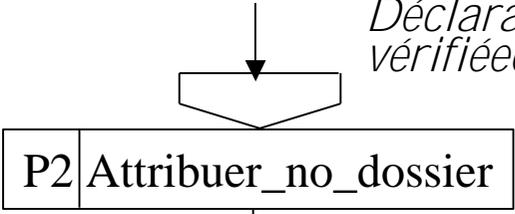
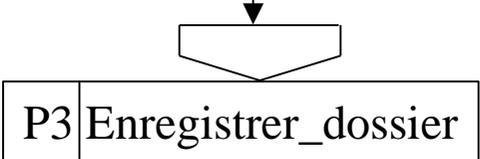
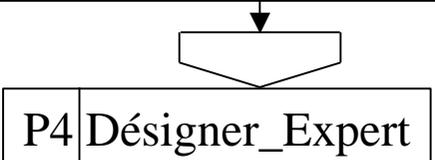
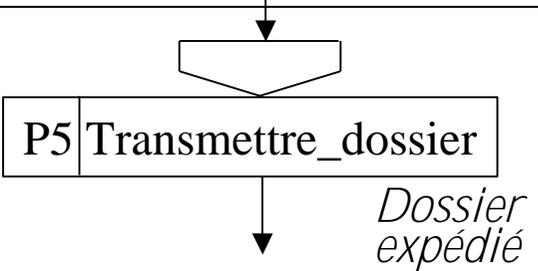
Alors *assuré_inconnu*

Sinon *déclaration_ok*

Fin

Enchaînement des procédures :

exemple

Date au plus tôt	Date au plus tard	Enchaînement des procédures
J	J	<p style="text-align: right;"><i>Déclaration vérifiée</i></p> 
J	J	
J	J + 2	 <p style="text-align: right;"><i>Dossier ouvert</i></p>
J	J + 3	 <p style="text-align: right;"><i>Dossier expédié</i></p>

Adaptation des modèles logiques

- Adapter les schémas logiques (SLD et SLT) dans le but de réduire le coût d'implantation de la base
- Facteurs à prendre en compte :
 - volume des données
 - nombre d'accès à la base
 - coût des traitements (supposé négligeable par rapport au coût des lectures - écritures dans la base)

Évaluation des volumes (1/2)

- Taille d'une relation représentant un type d'entité :
 - attribut : nombre de caractères nécessaires à sa représentation
 - n-uplet : somme des tailles de ses types d'attributs
 - relation : produit du nombre d'occurrences du type d'entité par la taille du n-uplet de la relation
- Exemple :

Attribut	Type	Taille
no_ass	entier	10
nom_ass	chaîne	30
adr_ass	chaîne	40
te_ass	chaîne	10
no_agence	entier	10

Si 10000 assurés sont attendus sur une période de deux ans, l'estimation de la taille de la relation est de
 $10000 \times (10 + 30 + 40 + 10 + 10) \approx 1\text{Mo}$

Evaluation des volumes (2/2)

- Taille d'une relation représentant un type d'association : suppose connu le nombre moyen d'occurrences des types d'entité associés
- Exemple :
 - " Un produit est stocké en moyenne dans 3 dépôts "
 - *taille de la relation stock = taille d'un n-uplet de stock × 3*
× nombre de produits

Optimisation des volumes

Seul type d'optimisation possible : compression des
types d'attributs

→ perte de lisibilité et coût supplémentaire dû au
processus de compression et de décompression

Évaluation du coût des traitements

- Dépend du type des opérations de base (recherche, insertion, suppression, modification)
- Processus d'évaluation et d'optimisation :
 1. évaluer le coût de chaque type d'opération (SLD et SLT → fréquence de chaque opération, objets concernés et actions élémentaires à effectuer sur ces objets)
 2. identifier les opérations les plus coûteuses
 3. essayer d'en réduire le coût

Coût de la recherche d'un n-uplet (1/2)

- La recherche d'un n-uplet dans un ensemble de n n-uplets coûte :
 - 1 accès si un mécanisme d'accès direct existe
 - en moyenne $n/2$ accès sinon

Coût de la recherche d'un n-uplet (2/2)

- Remarques :
 - coût d'un accès direct < coût d'un accès séquentiel
→ *créer des index*
 - coût d'une recherche dans un ensemble ordonné < coût d'une recherche dans un ensemble non ordonné
→ *ordonner les instances*
 - coût d'une lecture < coût de plusieurs lectures
→ *fusionner des relations, introduire de la redondance ou grouper physiquement des occurrences (clustering)*

Coût de l'ajout d'un n-uplet

- Insertion :
 - écriture dans la base
 - mise à jour éventuelle des index existant sur la relation concernée
- Coût :
 - 1 écriture si la relation n'est pas ordonnée
 - $n/2$ lectures en moyenne pour la recherche du point d'insertion si la relation est ordonnée

Coût de la modification d'un n-uplet

- Modification :
 - recherche dans la base
 - modification en mémoire centrale
 - réécriture dans la base
- Coût :
 - coût d'une recherche
 - coût d'un ajout
 - (éventuellement) coût du maintien d'ordre
 - (éventuellement) coût de mise à jour d'index
 - (éventuellement) coût de mise à jour de données redondantes

Coût de la suppression d'un n-uplet

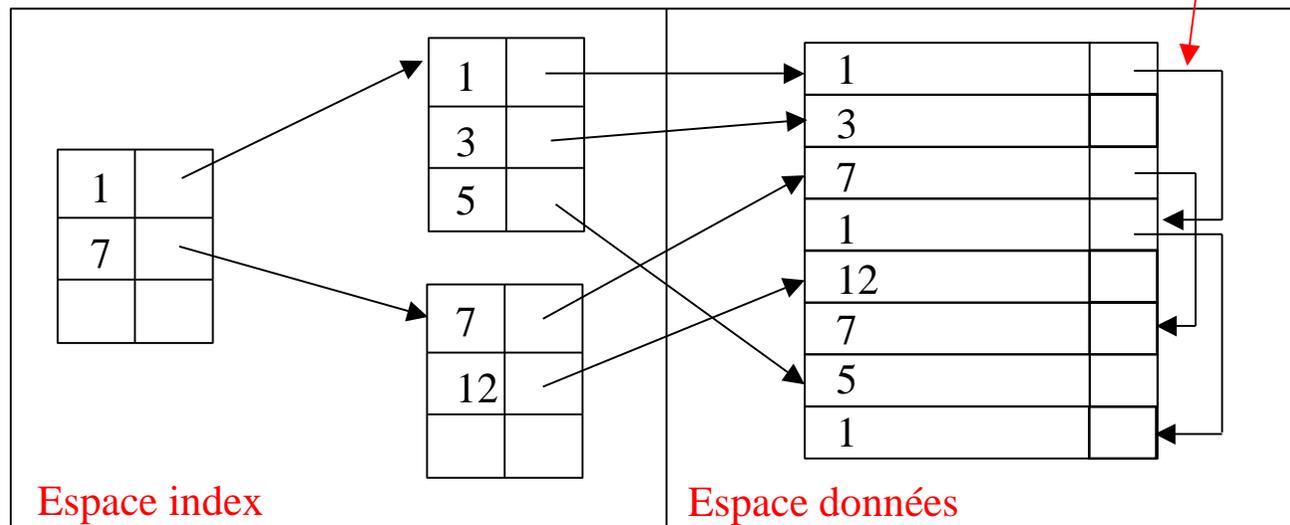
- Suppression :
 - recherche du tuple à supprimer
 - mise-à-jour éventuelle d'index
 - autres suppressions, dans le cas de données redondantes
- Coût :
 - coût d'une recherche
 - (éventuellement) coût de la maintenance des index
 - (éventuellement) coût de la suppression des données redondantes

Optimisation des traitements

- Index et critères d'ordre
- Redondance et dénormalisation de relations
- Ajout de nouveaux types d'attributs
- Fusion de relations
- Fragmentation verticale de relations

Rappel sur les index

- Index : structure de données qui associe à une valeur d'un attribut, appelé clé de l'index, la ou les adresses des tuples contenant cette valeur
- Exemple :



Choix des index et des critères d'ordre

- Opérations d'interrogation :
attributs de sélection et de jointure
- Opérations de mise à jour :
attributs de sélection
pas les attributs à modifier car cela entraînerait un coût supplémentaire pour la maintenance

Redondance et dénormalisation de relations

- Exemple :
Expert(no_exp, nom_exp, ...) / Expert chargé du dossier
Sinistre(no_dossier, ..., no_exp)
Si le nom de l'expert est accédé à chaque référence à un sinistre → Sinistre(no_dossier, ..., no_exp, nom_exp)
Viole la 3NF de Sinistre, mais permet de faire l'économie de l'opération de jointure pour obtenir le nom
→ accroissement de l'espace de stockage et de l'activité en mise à jour

Ajout de nouveaux types d'attributs

- Exemple :

" L'expert désigné pour un dossier de sinistre est celui qui a le moins de dossiers en cours d'instruction "

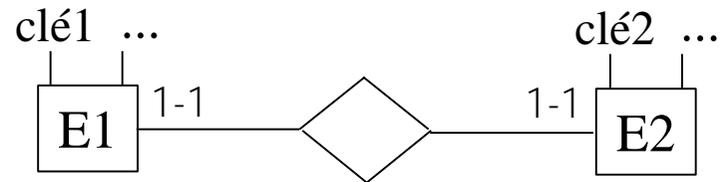
→ nécessite le parcours de la relation sinistre avec comptage et recherche du `no_exp` ayant le plus petit nombre de dossiers puis accès à la relation `Expert` pour connaître ses coordonnées

→ `Expert(no_exp, nom_exp, ..., nb_dossiers-en_cours)`

→ Attention à la cohérence de la base :

incrémenter `nb_dossiers-en_cours` à chaque fois que l'expert est désigné, et décrémenter quand le dossier est clos

Fusion de relations



RE1(clé1, ...)

RE2(clé2, ...)

→ si jointures très fréquentes de RE1 et RE2,
fusionner en une seule relation

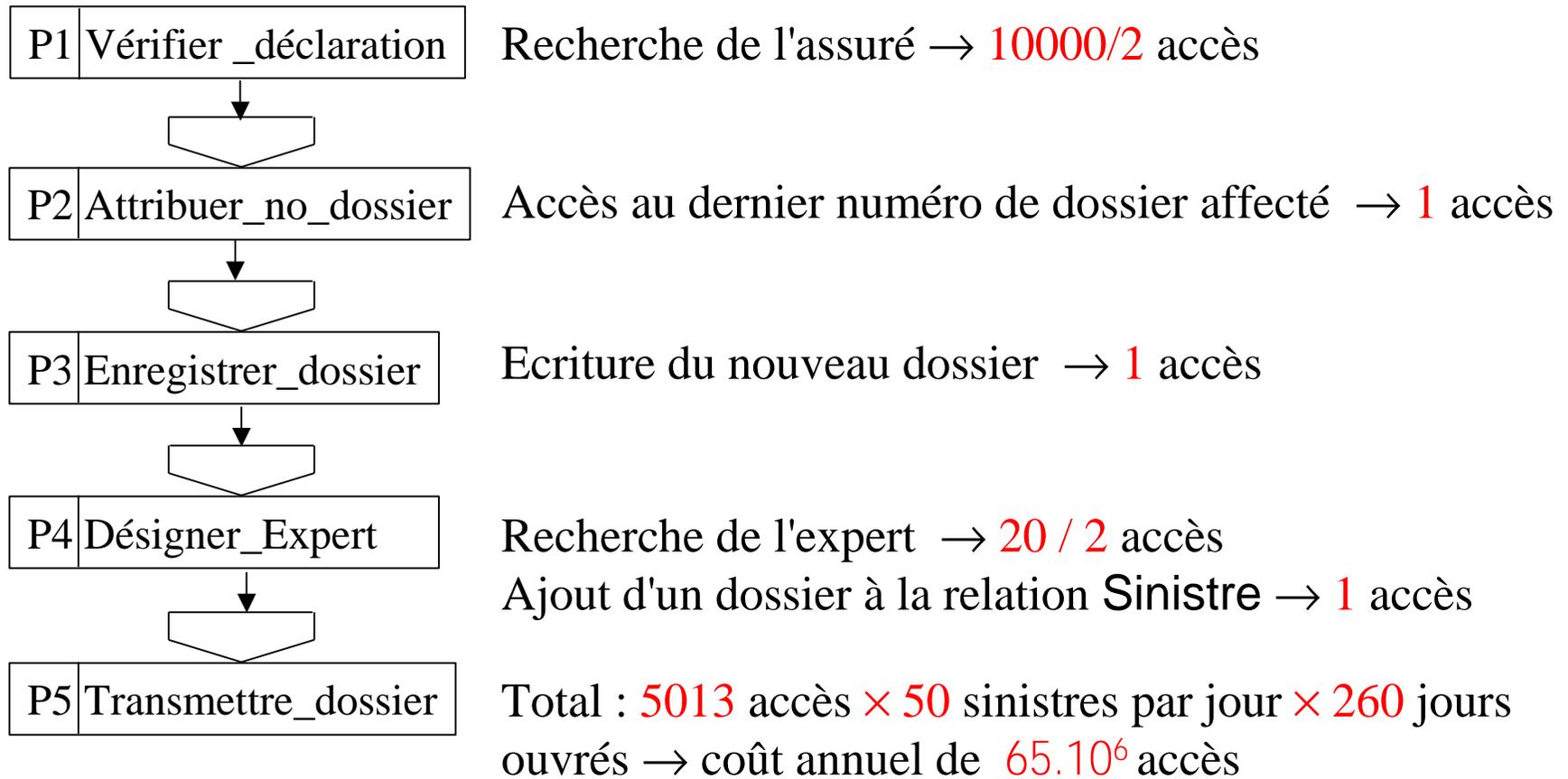
Fragmentation verticale de relations

- Quand une relation comporte un grand nombre d'attributs, dont seul un sous-ensemble est fréquemment utilisé, on peut la décomposer en deux relations
- Exemple :
R(cléR, utilisé1, utilisé2, peu_utilisé1, peu_utilisé2, peu_utilisé3)
→ R1(cléR, utilisé1, utilisé2)
R2(cléR, peu_utilisé1, peu_utilisé2, peu_utilisé3)

Exemple : coût de l'opération

Ouvrir_dossier

Pour 10000 dossiers et 20 experts :



Coût de l'opération

Ouvrir_dossier optimisée

- Si on décide de :
 - conserver en mémoire le dernier numéro de dossier pendant toute une journée de travail
 - indexer les assurés sur leur numéro de police
 - indexer les experts sur le nombre de dossiers en cours avec ordre

→ Nouveau coût :

$$260 \text{ jours} \times (1 \text{ accès au dernier numéro de dossier} + 50 \text{ sinistres} \times (1 \text{ pour la recherche de l'assuré} + 1 \text{ pour l'écriture du nouveau dossier} + 1 \text{ pour la recherche de l'expert} + 1 \text{ pour l'ajout d'un sinistre})) = \underline{52.10^3} \text{ accès / an}$$