Pour un système d'information

Module MED Bioinformatique et ontologies, *Université Montpellier 2, 4-6 mai 2010*

Patrice DUROUX

Patrice.Duroux@igh.cnrs.fr

Ingénieur de recherche

IMGT, Institut de Génétique Humaine, UPR 1142 CNRS

Généralités

Des bases de données...

Qu'est-ce qu'une base de données (BD) ?

- Sans informatique :
  - Une BD est consituée d'un ensemble de fichiers et des liens logiques entre ces fichiers.
  - Un fichier est une collection de fiches.
  - Une fiche comporte un ou plusieurs renseignements sur un élément (objet ou personne) de la BD.
  - Chacun de ces renseignements (appelés aussi champs) est une information indivisible.

Exemple : les fichiers tiroirs ou casiers des bibliothèques.

...Aux bases de données

- Avec informatique :
  - La définition d'une BD est identique.
  - Les concepts utilisés en informatique sont quelque peu différents.
  - Le modèle de représentation des données est considéré.

  Exemple : avec le modèle relationnel, une BD est définie par un ensemble de relations (des tables et des dépendances entre ces tables).

Historiquement

- Dès les années 60 : apparition des premiers systèmes d'exploitation (SE) pour faciliter la gestion physique des données (stockage magnétique, papier, ...)
- Puis les systèmes de gestion de fichiers (SGF) : séquentiel, relatif, indexé, séquentiel indexé, ... => indépendance «relative» des programmes et des données.
- Ensuite, les systèmes de gestion de bases de données (SGBD) :
  Une BD est, en général, créée pour être consultée. Elle peut être également mise à jour. Pour faciliter ces opérations de consultation et mise à jour, un outil logiciel est appréciable. Il existe aujourd'hui plusieurs logiciels de gestion de BD.

Apparition des SGBD

- Conférence «Development and Management of a computer-centered data base» (Santa Monica, 1964).
- Considération pour de gros volumes d'informations avec souci d'efficacité des accès.
- SGBD est un système de stockage des données et assure une facilité de maintenance.

Base $\neq$ Banque = index ou ensemble d'index (le référentiel versus le factuel).

Évolution du contexte informatique

- Matériel informatique : temps d'accès et de calcul $\searrow$, capacité mémoire $\nearrow$, coût $\searrow$.
- Systèmes et logiciels : utilisateurs multiples.
- Interconnexion et réseau : communication et disponibilité.
- Développement d'applications : génie logiciel, environnement intégré, modèle 4 Génération (MVC).

Système (en très bref)

Un *système* est un ensemble d'éléments (matériels ou pas) en «interaction» entre eux. Il comporte généralement des éléments en entrée et des éléments en sortie.
Pour nous, les systèmes seront constitués par des organisations (institution, entreprise, administration, collectif, etc.) et fonctionnent en vue de la réalisation d'objectifs déterminés.

Exemple 1 Une entreprise qui fabrique/commercialise des produits à :

- en entrée : des produits achetés, des commandes, les paiements des clients;
- en sortie : des produits vendus, des factures, les paiements aux fournisseurs.

Exemple 2 Un système algébrique d'équations.

Systèmes parmi les systèmes

On peut distinguer des systèmes les sous-systèmes connexes suivants :

- *système de pilotage* : régulation, contrôle, décision, définition d'objectifs ;
- *système opérationnel* (ou opérant) : réalisation d'actions ;
- *système d'information* (SI) : interface entre les deux précédents.

Système d'information

Composé d'éléments divers (personnels, ordinateurs, conventions, ...), le SI informe le système de pilotage sur les fonctionnement du système opérant, et renvoie au système opérant des directives provenant du système de pilotage.
Le SI est la «mémoire» de l'organisation avec 2 aspects :

- statique : enregistrer des faits, des règles et des contraintes.
- dynamique : mettre à jour ses enregistrements.

Nature de l'information

Le SI d'une organisation regroupe tout ce qui, à quelque niveau que se soit, traite ou stocke ses informations relatives :

- aux flux : produits en stocks, produits commandés, bons de livraison, factures, bons de commandes...
- à l'univers extérieur : clients, fournisseurs...
- à l'organisation de l'entreprise : que se passe-t-il entre l'enregistrement d'une commande et sa livraison ?
- aux contraintes légales : lois, règlements, paramètres financiers...
- *etc.*

Qu'est-ce qui peut être informatisé ?

D'une manière générale, seule une partie du SI peut-être automatisée, c'est ce que l'on appelle *système automatisé d'information* (SAI).
Le SAI communique avec son environnement extérieur par des saisies et des accès. Il contient une partie mémorisation et une partie traitement.

- Mémorisation : stockage des données, structuration des données, règles de fonctionnement.
- Traitement : contrôle des données, mise à jour, recherches, calculs.
- Interface entre l'extérieur (univers) et le SAI : saisie, accès.

Concevoir un SI

De nos jours l'informatique est tant technologie que discipline pour les SI.
L'analyse (informatique) d'un SI aboutie traditionnellement aux modèles :

- de données (MDD), la formalisation des données (informations de toute nature) présentes à un moment ou un autre dans le système;
- de traitement (MDT), la formalisation des traitements (i.e des processus dynamiques) intervenant dans le système.

Périmètre du formalisable

- Données formalisables : renseignements sur un client, une commande, les cours de la bourse, la température et pression.
- Données non formalisables : conjoncture économique, météo, rendement d'un individu (frontière floue).
- Traitements formalisables : édition d'un facture, renouvellement des stocks, lettre de rappel.
- Traitements non formalisables : si le fournisseur ne donne pas satisfaction alors en changer, arrêter la fabrication d'un produit et en créer un autre.

Difficultés

Le problème du traitement des choix :

- rarement automatisables, du ressort de l'être humain;
- sauf cas particuliers, qu'il est possible de les formaliser dans le modèle.

Exemple : le renouvellement d'un stock sera du ressort de l'être humain, sauf si on peut le formaliser par un règle comme «si stock(xx)<..., alors commander(xx,yy)».

Outils de modélisation

MERISE : Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise

- Fruit d'une consultation nationale du Ministère (fin années 70).
- Méthodologie : modèles conceptuel, logique et physique.
- Approche données/traitements.

UML : Unified Modeling Language

- Factorisation d'approche existantes : OMT, Booch et OOSE (années 90).
- Notation (type de diagrammes) et recommandation.
- Approche dite «objets».

Le cas d'IMGT

Infrastructure

Répartie :

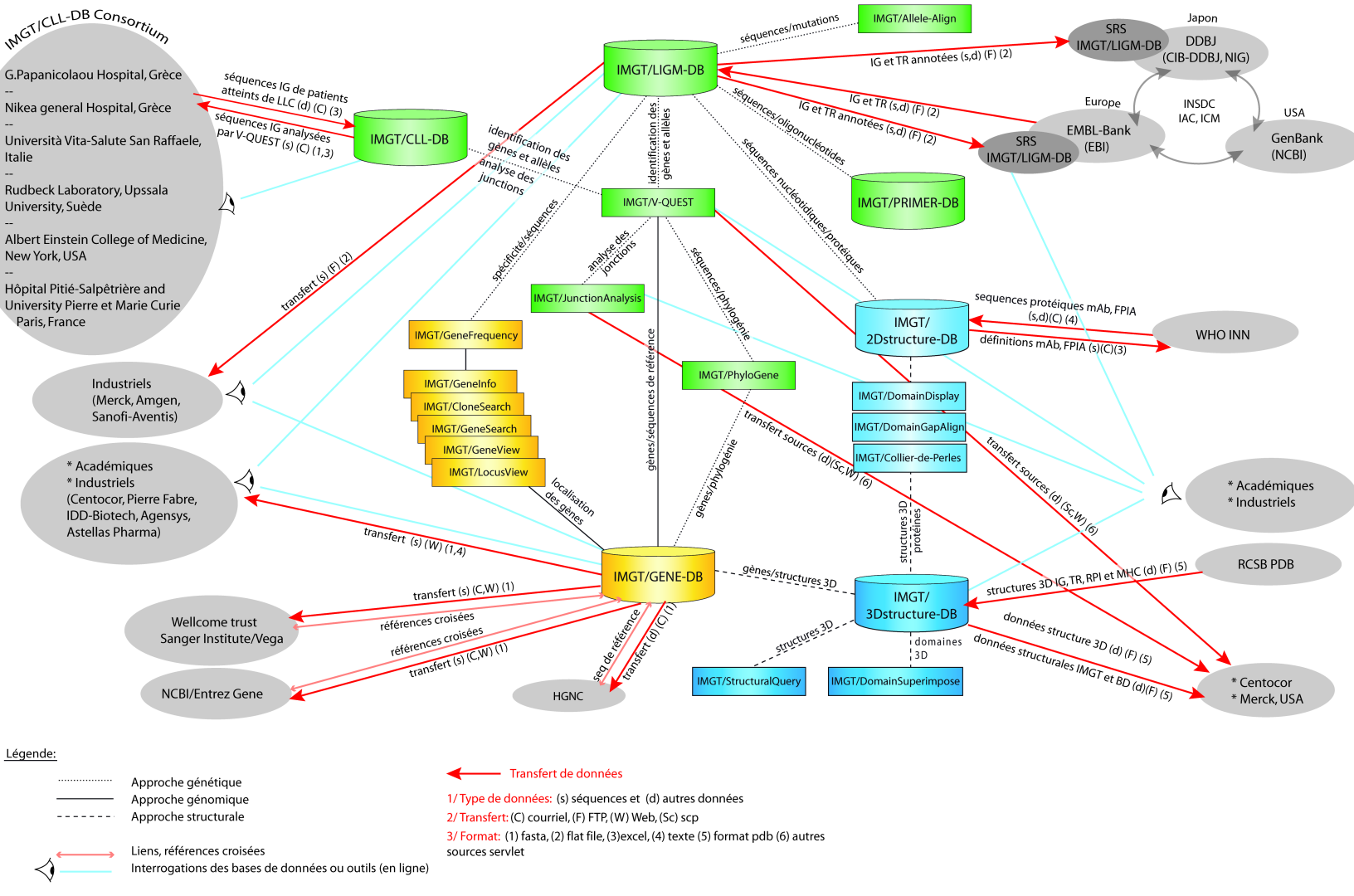
- personnel sur 2 sites (IGH et UM2) : management, informaticiens, bio-informaticiens, bio-curateurs.
- informatique sur 3 sites (IGH, UM2 et CINES) : 10 serveurs + 15 postes PC.

Hétérogène :

- compétence pluri-disciplinaire
- systèmes informatiques variés : SE (Windows, Linux), SGBD (Sybase, MySQL), langages de programmation (Shell, Perl, Java, PHP)
- formats bio-informatiques : texte «brut» ou tabulée (CSV), données séquences et structures (FASTA, EMBL, PDB), documents (HTML, PDF), diagrammes (bitmap, SVG)

Composants et relations

Relations entre IMGT® et ses différents collaborateurs et utilisateurs.

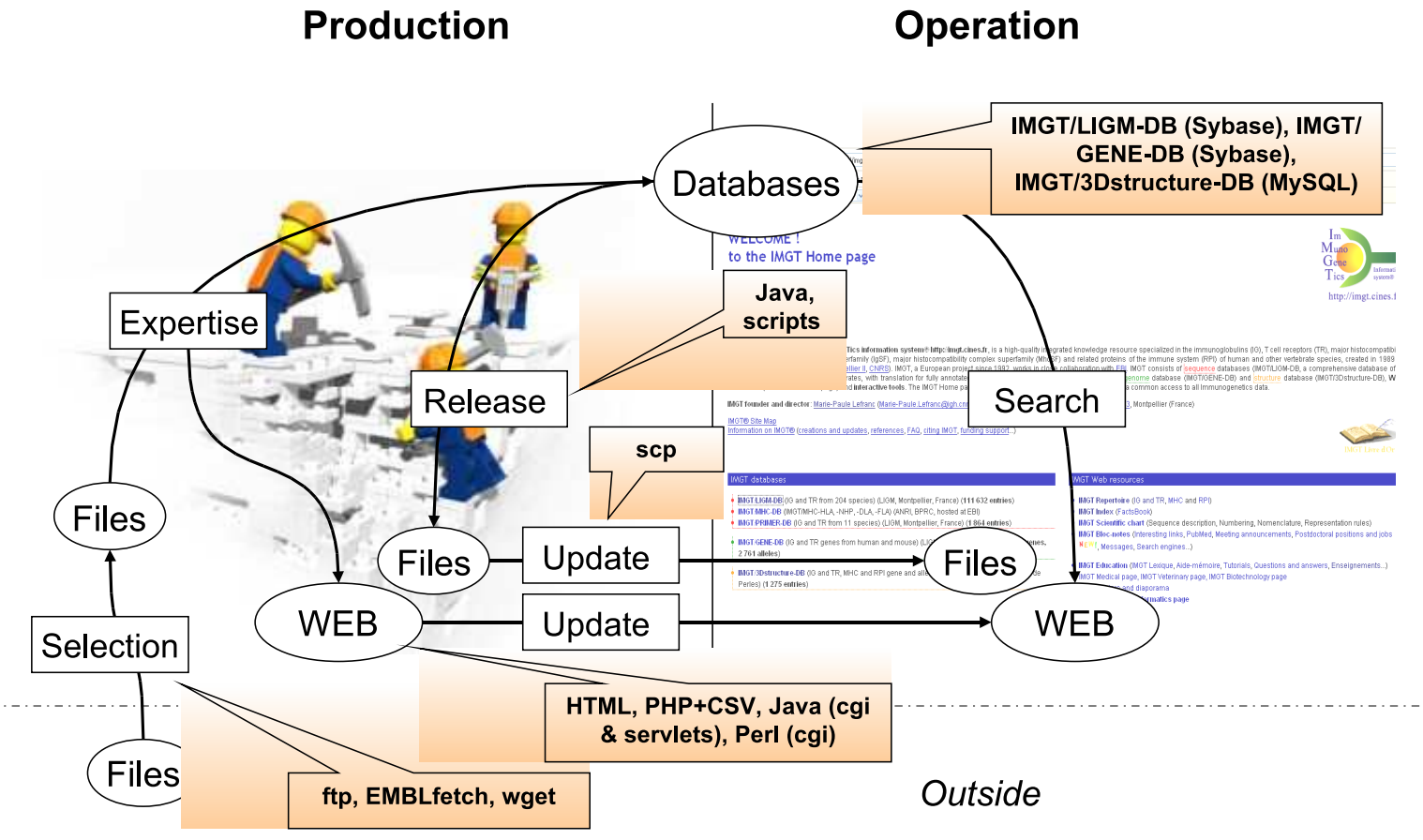


Objectifs

- Maintenir une cohérence scientifique
- Garantir une expertise biologique adéquate
- Faciliter les échanges
- Gérer les dépendances
- Organiser les révisions
- Obtenir une traçabilité
- Introduire des indicateurs et statistiques
- S'orienter vers une «démarche qualité»

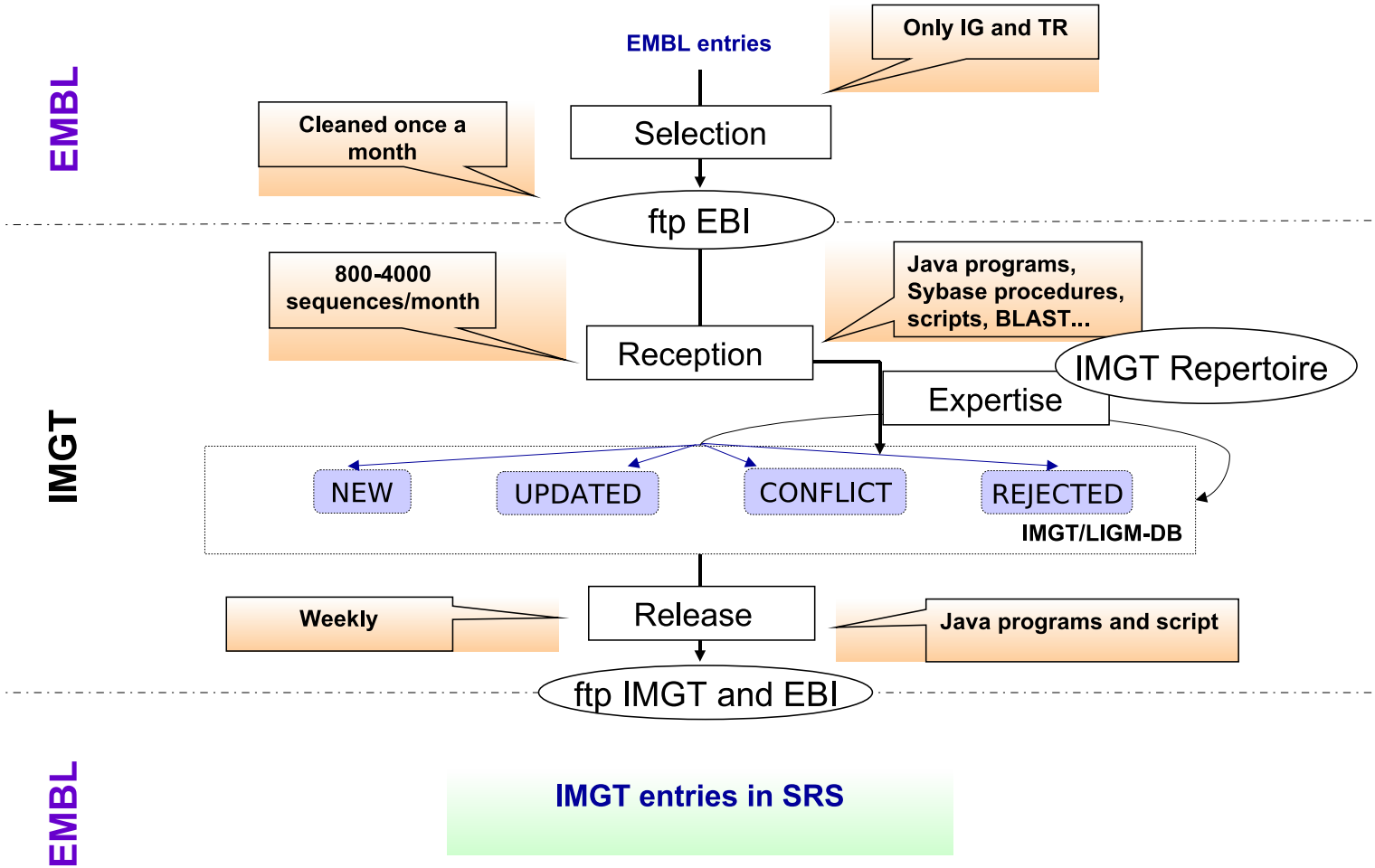
IMGT data-flow

Data-processing infrastructure within IMGT®

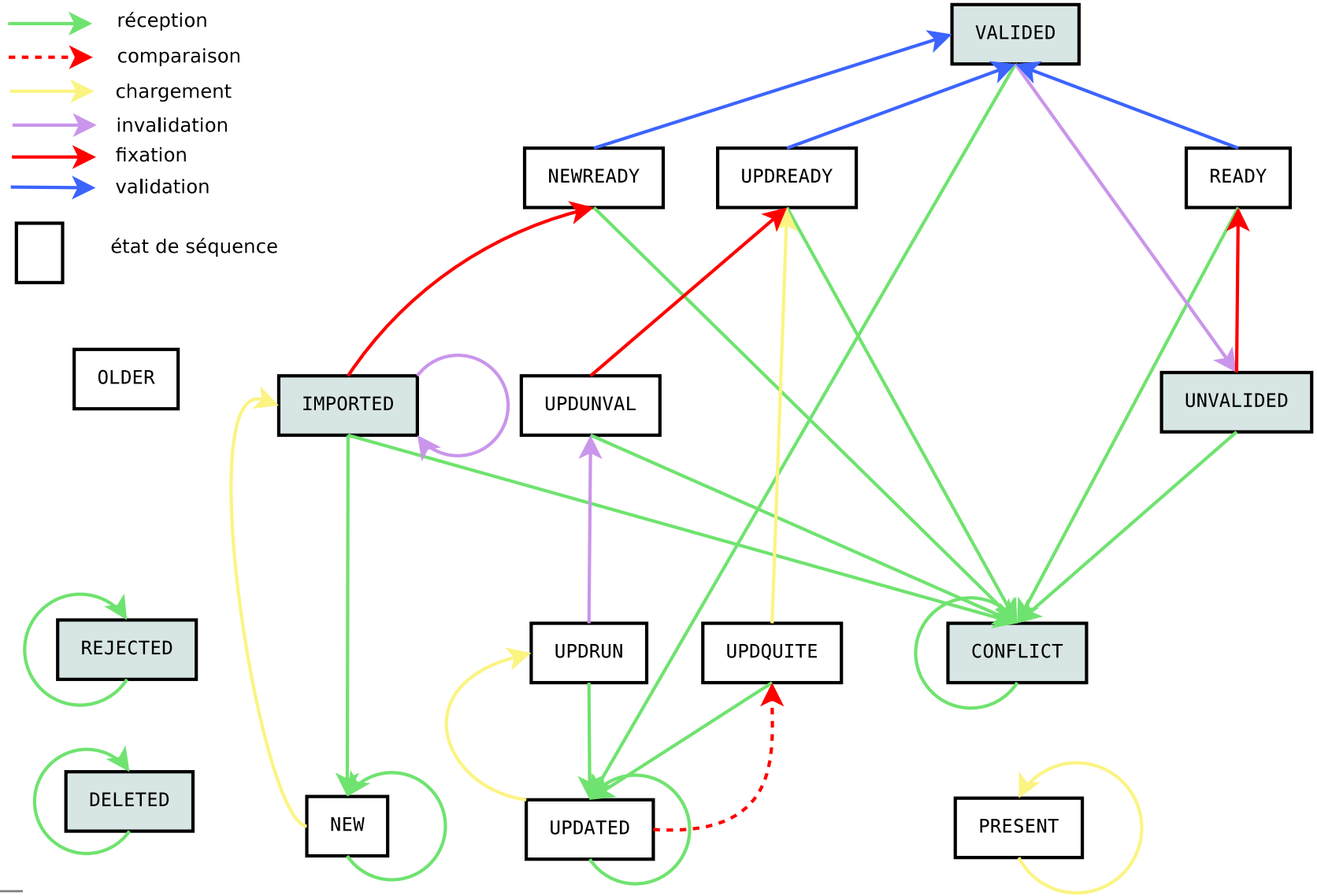


IMGT/LIGM-DB data-flow

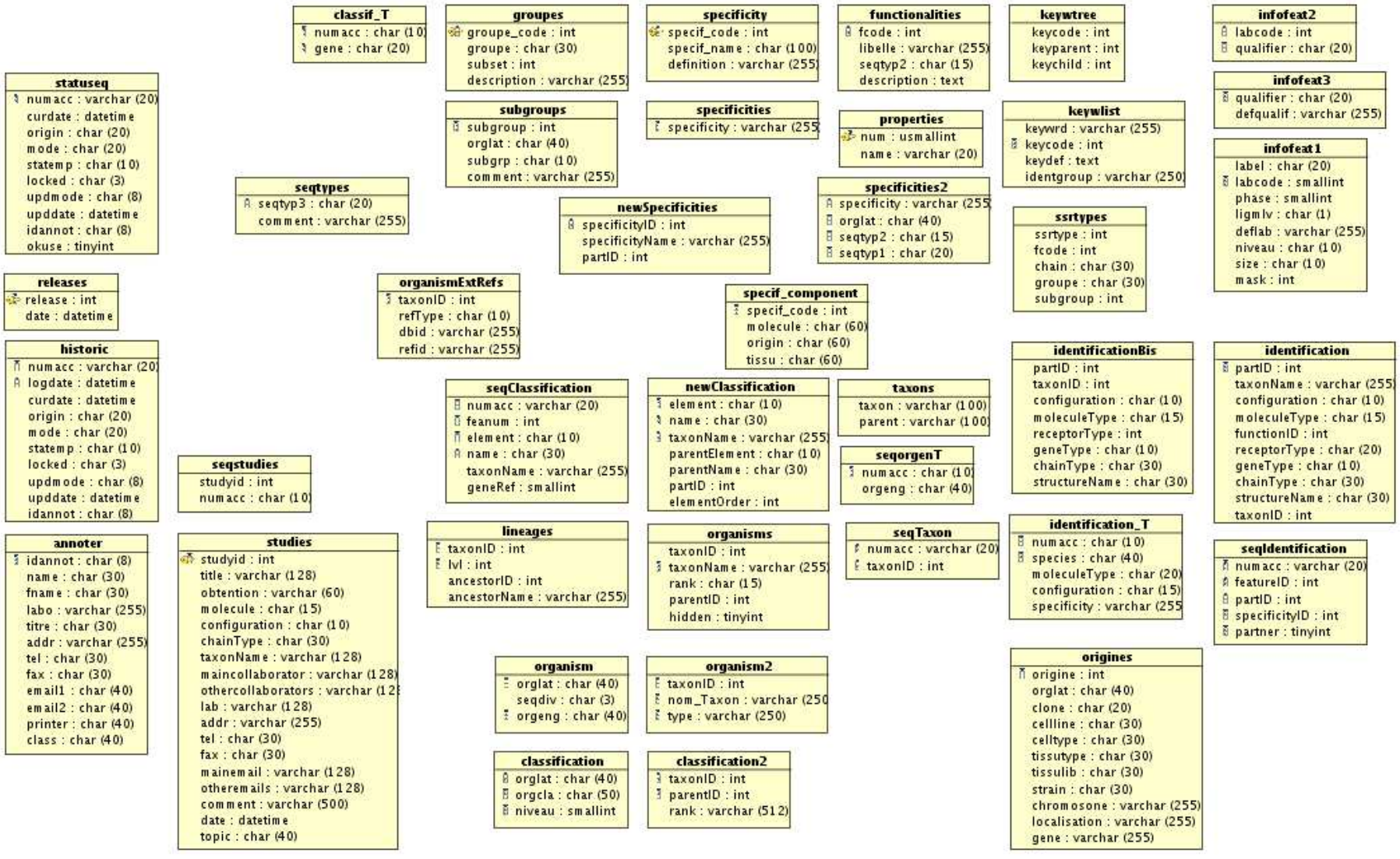
IMGT/LIGM-DB (with EMBL)



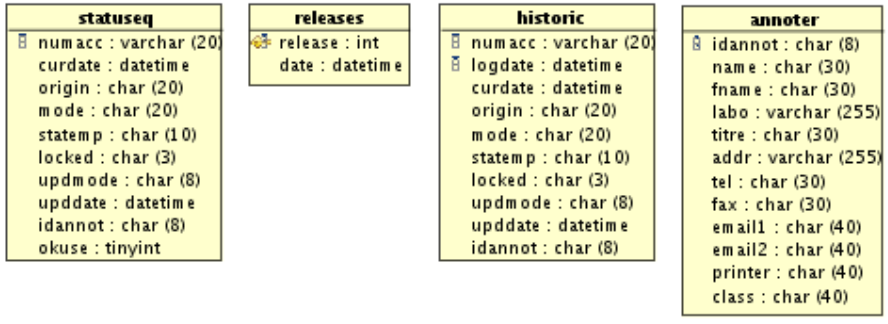
Etat des séquences (IMGT/LIGM-DB)



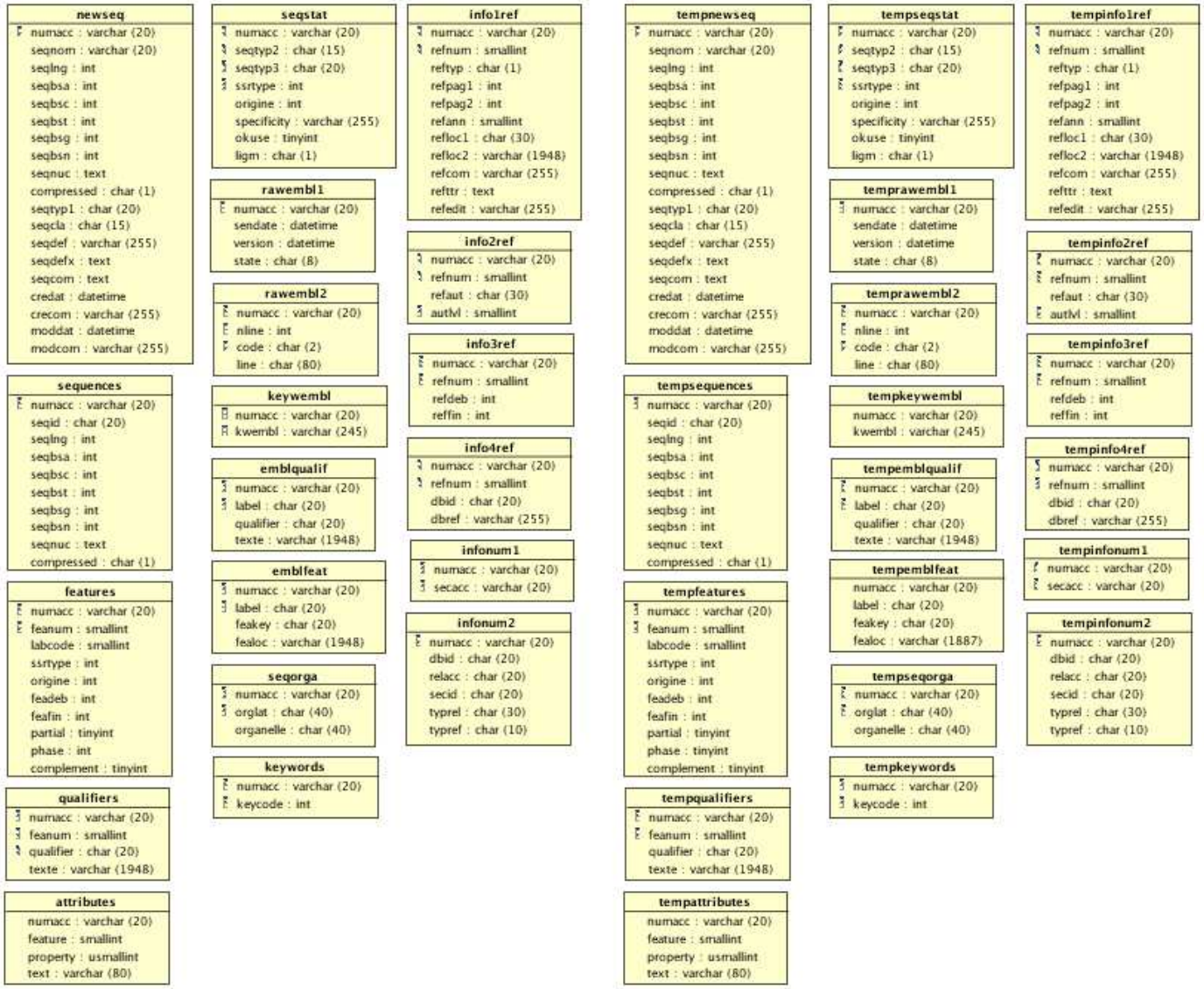
Modèle des connaissances



Modèle des méta-données



Modèle des séquences



Algèbre relationnelle

Introduction

Historique

1960 Systèmes de gestion de fichiers/fiches

1970 Systèmes hiérarchiques et réseaux

1980 Systèmes relationnels commerciaux

1990 Systèmes à objets

3 grands leaders des systèmes relationnels commerciaux : Oracle, Informix et Sybase. Mais aussi IBM avec DB2 et Microsoft avec Access.

Introduction

Motivations

Edgar F. CODD (1970) : définir une algèbre pour formaliser mathématiquement les traitements applicables à des bases de données $\Longrightarrow$ modèle de base de données relationnelle.

- Modèle simple avec peu de concepts et facile à appréhender pour "percevoir" et "implanter" les données.
- Représentation de l'information dans un ensemble de relations (la base).
- Langage commun de définition et de manipulation des données : le SQL.
- Théorie mathématique sous-jacente à ce langage.

Introduction

Références bibliographiques

- Georges Gardarin, *Bases de données*, Eyrolles, 1999.
- Jean-Luc Hainaut, *Bases de données et modèles de calcul*, Dunod, 2000.
- Jeffrey D. Ullman, *Principles of Database and Knowledge-Base Systems. Volume I*, Computer Science Press, 1988.

Exemple

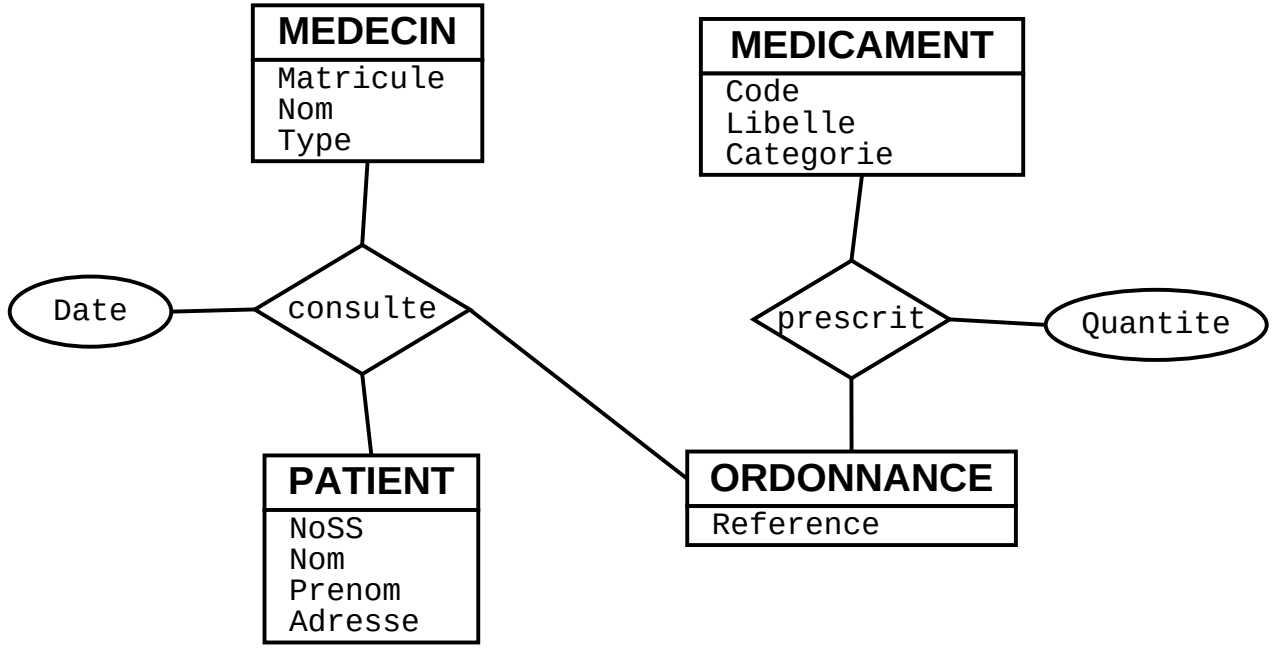
Problématique

Traiter l'information relative à un centre médical.
Notamment, pouvoir retrouver des données en répondant à des questions (dites *requêtes*) comme :

- Quel est le numéro de sécurité sociale de M. Amil ?
- Quel(s) médecin(s) a-t-il consulté ?
- Quelle est l'ordonnance de sa dernière consultation ?
- Qui avait-il de prescrit ?
- ...

Exemple

Entités/Associations



Remarque : diagramme brut sans cardinalité, ni clef.

Exemple

Tables

MEDECIN

Matricule	Nom	Type
271	Dubois	Généraliste
865	Malchausse	Dentiste

PATIENT

NoSS	Nom	Prenom	Adresse
1720833245080	Amil	Jérôme	Nîmes
2740284123456	Brand	Emilie	Nîmes

Exemple

Schéma relationnel

Entités $\rightarrow$
- MEDECIN(Matricule,Nom,Type)
- PATIENT(NoSS,Nom,Prenom,Adresse)
- ACTE(Reference,Date)
- MEDICAMENT(Code,Libelle,Categorie)

Associations $\rightarrow$
- CONSULTE(Reference,Matricule,NoSS)
- PRESCRIT(Reference,Code,Quantite)

Remarque : ORDONNANCE = ACTE $\bowtie$ CONSULTE, soit ORDONNANCE(No,Matricule,NoSS,Date).

Modèle Relationnel

Intuitivement

- Fondée sur la théorie des ensembles.
- Relation : représente une collection (ensemble ou multi-ensemble) de données homomorphes (comme dans une table, voire un fichier = collection de fiches).
- Attribut : semblable à une colonne de la table, décrit un domaine.
- N-uplet : semblable à une ligne de la table, décrit une donnée.
- Algèbre : structure d'ensemble de relations muni de lois de composition interne $\Longrightarrow$ langage et égalité.

Langage Relationnel

Attributs

Exemple : Nom, Prenom, DateNaissance, ...
Un symbole parmi un ensemble $\mathcal{A} = \{A_1, \ldots, A_m, \ldots\}$

Domaines

Exemple : texte, nombre, date, ...
Un symbole parmi un ensemble $\mathcal{D} = \{D_1, \ldots, D_n, \ldots\}$ souvent fini et prédéfini (suivant les types de données retenus pour le système)
Chaque domaine D_i est un ensemble (fini ou non) de valeurs, ex. "Jérôme", 12, 10-01-2002, etc.
Remarque : les D_i sont non nécessairement disjoints.

Langage Relationnel

Relations

- Deux formes : intention (variable) et extension (n-uplet)
- Dénotent des collections de données
- Chaque relation est définie sur un *schéma* qui est l'ensemble des attributs qui les composent
- La *dimension* d'une relation est le cardinal de son schéma
- Base de données relationnelles : ensemble de relations

Relations

Variables

- Syntaxe : R
- Déclaration : $R(A_1 : D_1, \ldots, A_n : D_n)$
- Condition : les A_i tous distincts deux à deux ($A_i \in \mathcal{A}$)
- Sémantique : dénote un sous-ensemble de $D_1 \times \cdots \times D_n$ ($D_i \in \mathcal{D}$)
- Schéma : $R^+ = \{A_1, \ldots, A_n\}$
- Composante : $dom_R(A_i) = D_i$

Exemple : MEDECIN, PATIENT, ORDONNANCE, etc.

Relations

N-uplets

- Arité : nulle
- Syntaxe : $(A_1 : v_1, \ldots, A_n : v_n)$
- Condition : pour les $A_i \in \mathcal{A}$, les $v_i \in D_j$
- Sémantique : dénote une relation atomique ayant pour n-uplets $\{t\}$
- Schéma : $t^+ = \{A_1, \ldots, A_n\}$

Exemple : (NoSS: 1720833245080, Nom: Amil, Prenom: Jérôme, Adresse: Nîmes)

Propriétés

R-compatibilité : un n-uplet $(A_1 : v_1, \ldots, A_n : v_n)$ et une relation R ont le même schéma et les $v_i \in dom_R(A_i)$

Exemple : (NoSS: 1720833245080, Nom: Amil, Prenom: Jérôme, Adresse: Nîmes) est PATIENT-compatible

Schéma-compatibilité : deux relations qui ont le même schéma et les même domaines associés

Remarque : l'ordre des composants d'un n-uplet n'est pas important avec la notion d'attribut

Relations à attributs $\Longrightarrow$ relations classiques

Langage Relationnel

Opérations

- l'union $\cup$,
- l'intersection $\cap$,
- la différence $\setminus$,
- le produit cartésien $\times$.
- le renommage ρ,
- la projection π,
- la sélection σ,
- les jointures $\bowtie$.

Remarque : certaines usuelles sur les ensembles, d'autres plus spécifiques

Union

Formellement

- Arité : binaire
- Syntaxe : $R_1 \cup R_2$
- Condition : R_1 et R_2 schéma-compatible
- Sémantique : $R_1 \cup R_2 = \{t | t \in R_1 \vee t \in R_2\}$
- Schéma : $(R_1 \cup R_2)^+ = R_1^+ = R_2^+$

Intersection

Formellement

- Arité : binaire
- Syntaxe : $R_1 \cap R_2$
- Condition : R_1 et R_2 schéma-compatible
- Sémantique : $R_1 \cap R_2 = \{t | t \in R_1 \wedge t \in R_2\}$
- Schéma : $(R_1 \cap R_2)^+ = R_1^+ = R_2^+$

Différence

Formellement

- Arité : binaire
- Syntaxe : $R_1 \setminus R_2$
- Condition : R_1 et R_2 schéma-compatible
- Sémantique : $R_1 \setminus R_2 = \{t | t \in R_1 \wedge t \notin R_2\}$
- Schéma : $(R_1 \setminus R_2)^+ = R_1^+ = R_2^+$

Renommage (I)

Formellement

- Arité : unaire
- Syntaxe : $\rho_X(R)$
- Condition : $X = \{\dots A_i/B_i \dots\}$ où $A_i \in R^+$ et $B_i \in \mathcal{A}$
- Sémantique : $\rho_{A_1/B_1,\dots,A_n/B_n}(R) = \{(\rho_(A_1) : v_1, \dots, \rho_(A_n) : v_n) | (A_1 : v_1, \dots A_n : v_n) \in R\}$ où $\rho_X(A_i) = B_i$ si $(A_i/B_i) \in X$, $\rho_X(A_i) = A_i$ sinon
- Schéma : $(\rho_X(R))^+ = \{\rho_X(A_1), \dots, \rho_X(A_n)\}$
- Remarque : existe aussi avec un renommage de la relation $\rho_{R_2}(R_1)$

Renommage (II)

Exemple

$$\rho_{Nom/NomPatient}(\text{PATIENT})$$

NoSS	NomPatient	Prenom	Adresse
1720833245080	Amil	Jérôme	Nîmes
2740284123456	Brand	Emilie	Nîmes

Projection (I)

Intuitivement

Extraire des colonnes

Formellement

- Arité : unaire
- Syntaxe : $\pi_X(R)$ ou $R[X]$
- Condition : $X \subseteq R^+$
- Sémantique : $\pi_X(R) = \{(\ldots, A_i : v_i, \ldots) | (A_1 : v_1, \ldots, A_n : v_n) \in R \wedge A_i \in X\}$
- Schéma : $(\pi_X(R))^+ = X$
- Remarque : $|(\pi_X(R))^+| \leq |R^+|$

Projection (II)

Exemple

$\pi_{NoSS,Nom}(\text{PATIENT})$

NoSS	Nom
1720833245080	Amil
2740284123456	Brand

Sélection (I)

Intuitivement

Extraire des lignes

Formellement

- Arité : unaire
- Syntaxe : $\sigma_F(R)$ ou $R\{F\}$
- Condition : F est une formule logique sur le R^+ telle que pour tout $t \in R$, $F(t)$ est vrai ou faux
- Sémantique : $\sigma_F(R) = \{t | t \in R \wedge F(x)\}$
- Schéma : $(\sigma_F(R))^+ = R^+$
- Remarque : $\sigma_F(R) \subseteq R$

Sélection (I)

Exemple

$$\sigma_{Nom=Amil}(\text{PATIENT})$$

NoSS	Nom	Prenom	Adresse
1720833245080	Amil	Jérôme	Nîmes

Jointures (I)

Produit (cartésien)

- Arité : binaire
- Syntaxe : $R_1 \times R_2$
- Condition : $R_1^+ \cap R_2^+ = \emptyset$ sinon renommage préalable des attributs communs de R_1 et R_2 (préfixés par la relation)
- Sémantique : $R_1 \times R_2 = \{(A_1 : v_1, \ldots, A_m : v_m, B_1 : w_1, \ldots, B_n : w_n) | (A_1 : v_1, \ldots, A_m : v_m) \in R_1 \wedge (B_1 : w_1, \ldots, B_n : w_n) \in R_2\}$
- Schéma : $(R_1 \times R_2)^+ = R_1^+ \cup R_2^+$

Jointures (II)

Exemple

PATIENT2 $\times$ MEDECIN

NoSS	NomP	Prenom	Adresse	Matricule	Nom	Type
1720833245080	Amil	Jérôme	Nîmes	271	Dubois	Généraliste
1720833245080	Amil	Jérôme	Nîmes	865	Malchausse	Dentiste
2740284123456	Brand	Emilie	Nîmes	271	Dubois	Généraliste
2740284123456	Brand	Emilie	Nîmes	865	Malchausse	Dentiste

Jointures (III)

Theta-Jointure

- Arité : binaire
- Syntaxe : $R_1 \bowtie_\Theta R_2$
- Condition : Θ est une condition
- Sémantique : $R_1 \bowtie_\Theta R_2 = \{t | t \in R_1 \times R_2 \wedge \Theta(t)\}$
- Schéma : $(R_1 \bowtie_\Theta R_2)^+ = R_1^+ \cup R_2^+$
- Remarque : si $R_1^+ \cap R_2^+ =$ alors on a $R_1 \bowtie_\Theta R_2 = R_1 \times R_2$

Remarque : équi-jointure si Θ de la forme $R_1.A = R_2.B$

Jointures (IV)

Jointure naturelle

- Arité : binaire
- Syntaxe : $R_1 \bowtie R_2$
- Sémantique : dénote une équi-jointure avec une égalité entre paires d'attributs communs de R_1 et R_2 (équi-jointure dite naturelle)
- Remarque : cf. Theta-jointure

Égalités

Expriment des propriétés des opérateurs algébriques comme par exemple la commutativité du produit

- $R \times S = S \times R$
- $R \times (S \times T) = (R \times S) \times T$
- $\pi_Y(\pi_X(R)) = \pi_Y(R)$ si $Y \subseteq X$
- $\pi_Z(R \times S) = \pi_X(R) \times \pi_Y(S)$ où $X = Z \cap R^+$ et $Y = Z \cap S^+$
- $\pi_X(R \cup S) = \pi_X(R) \cup \pi_X(S)$

Égalités

- $\sigma_{F_1}(\sigma_{F_2}(R)) = \sigma_{F_1 \wedge F_2}(R)$
- $\sigma_F(R \cup S) = \sigma_F(R) \cup \sigma_F(S)$
- $\sigma_F(R \setminus S) = \sigma_F(R) \setminus \sigma_F(S)$
- $\sigma_F(R_1 \times R_2) = \sigma_{F_1}(R_1) \times \sigma_{F_2}(R_2)$ où $F = F_1 \wedge F_2$ et F_i porte sur R_i (inclus le cas vide pour un F_i)
- $\pi_X(\sigma_F(R)) = \sigma_F(\pi_X(R))$ si F porte sur X sinon $\pi_X(\sigma_F(R)) = \pi_X(\sigma_C(\pi_{YX}(R)))$ où $Y = var(F) \setminus X$

Langage Relationnel

Résumé

- Chaque opération détermine une relation (clos)
- Elles peuvent être composées (liberté)

$\Longrightarrow$ détermine une Algèbre dite Relationnelle

- Terme (expression du langage algébrique) :

$$T \doteq R|t|(T \cup T)|(T \cap T)|(T \setminus T)$$

$$|\rho(T)|\pi_X(T)|\sigma_F(T)|(T \times T)|(T \bowtie T)$$

- Requêtes = termes de l'algèbre relationnelle

Vers SQL

Intuitivement :

$$\pi_X(\sigma_F(R)) \Longleftrightarrow \begin{cases} \texttt{select}\ X \\ \texttt{from}\ R \\ \texttt{where}\ F \end{cases}$$

à des transformations des termes X, R et F près.
Exemple : si en SQL, $R = R_1, \ldots, R_l$, alors en Algèbre Relationnelle $R = \rho_?(R_1) \times \cdots \times \rho_?(R_l)$

Vers SQL

- Extension sémantique : ensembles $\rightarrow$ multi-ensembles
- Opérations étendues : classiques + projection, sélection, jointure
- Remarques : certaines propriétés sont perdues, ex. $R \cap (S \cup T) \neq (R \cap S) \cup (R \cap T)$
- Opérations supplémentaires :
  - élimination des duplicatas δ,
  - trie $\tau_X(R)$ (selon les attributs de X),
  - agrégation $\phi(X)$ avec ϕ somme, moyenne, cardinal, min, max, etc.
  - groupement $\gamma_X(R)$ (avec agrégations).

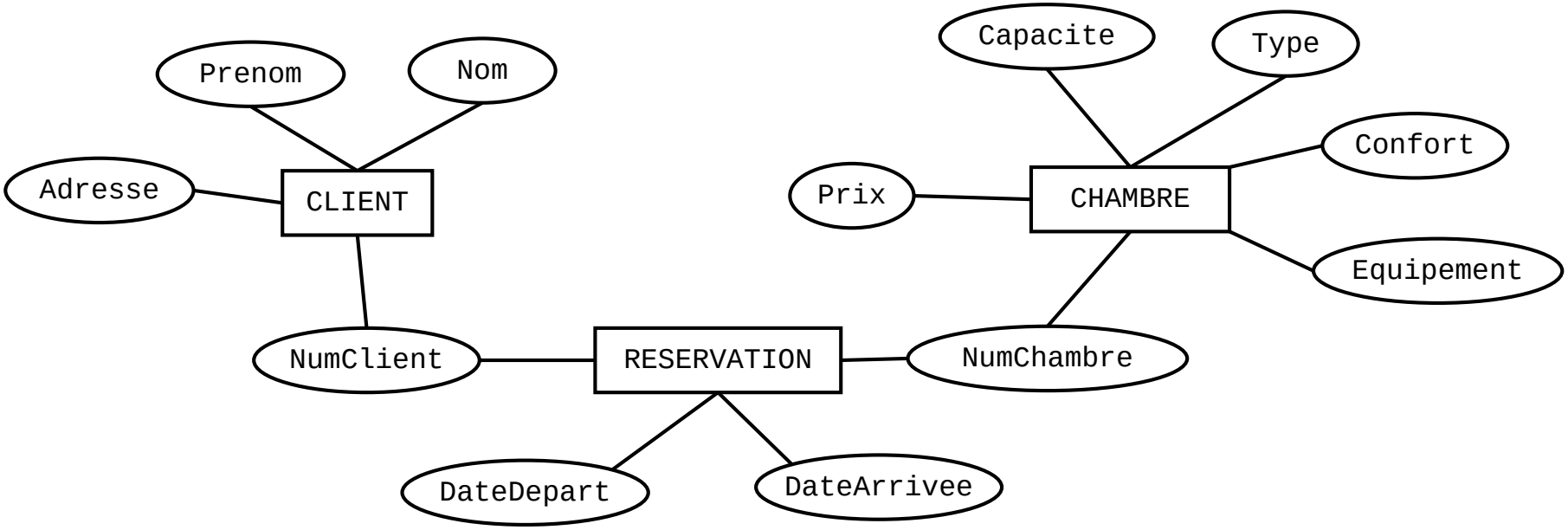
Conclusion

Conséquences

- Plusieurs algèbres équivalentes suivants le choix des primitives (opérations de base)
  - $R \cap S = R \setminus (R \setminus S)$
  - $R \bowtie_\Theta S = \sigma_\Theta(R \times S)$
  - $R \bowtie S = \pi_?(R \bowtie_? S)$
- Algèbre "standard" : SPC (Sélection/Projection/produit Cartésien)
- Facilité la formulation et/ou l'évaluation des requêtes $\Longrightarrow$ optimisations
- Langage trop libre $\Longrightarrow$ contraintes

Exercice

Chambres d'hôtel



Exercice

Soient les questions :

1. Les chambres avec bain et télévision ?
2. Les numéros des chambres et leur capacité ?
3. Les noms de clients ayant réservés une chambre pour le 25/12/2001 ?
4. Le nom des clients et le confort des chambres qu'ils ont réservés ?
5. La capacité théorique d'accueil de l'hôtel ?

Exprimer ces requêtes en algèbre relationnelle.

Modèle Relationnel

- Objectif : éviter les inaptitudes/répétitions/pertes d'information.
- Incovénient : système AR trop libre $\Longrightarrow$ ajouter des formes de contraintes.
- Problème : comparer plusieurs schémas relationnels équivalents pour une même base (ensemble de données) car certains sont plus «efficaces» que d'autres pour l'ajout, la modification et la suppression.
- Une solution : ajouter des dépendances fonctionnelles au schéma

Exemple

PATIENT

NoSS	Nom	Prenom	Adresse
1720833245080	Amil	Jérôme	Nîmes
2740284123456	Brand	Emilie	Nîmes

Est-il possible d'ajouter le n-uplet (NoSS : 1720833245080, Nom : Augier, Prenom : Rémi, Adresse : Montpellier) ?

Oui ? Non ? Notion d'attributs identifiants ou clefs (ici NoSS)

Dépendances Fonctionnelles

Informellement

Dépendance fonctionnelle : attributs qui déterminent d'autres attributs

Sur-clef : cas particulier où tous les attributs sont déterminés

Clef : plus petit ensemble pour l'inclusion (minimalité)

Notation : $\underline{A}$ désigne un attribut de la clef

Exemple

Schéma relationnel

Entités $\rightarrow$
- MEDECIN(Matricule,Nom,Type)
- PATIENT(NoSS,Nom,Prenom,Adresse)
- ACTE(Reference,Date)
- MEDICAMENT(Code,Libelle,Categorie)

Associations $\rightarrow$
- CONSULTE(Reference,Matricule,NoSS)
- PRESCRIT(Reference,Code,Quantite)

Dépendances Fonctionnelles

Cadre

Unique relation R sur les attributs $\mathcal{A}$ (Schéma *universel* d'une base)

Langage

- Syntaxe : $X \rightarrow Y$ où X et Y sont des parties de $\mathcal{A}$
- Sémantique : pour toute instance de R et pour tous n-uplets t_1 et t_2 tels que $\pi_X(t_1) = \pi_X(t_2)$, on a : $\pi_Y(t_1) = \pi_Y(t_2)$

Dépendances Fonctionnelles

Modèle

- si $XY = \mathcal{A}$ alors deux n-uplets t et t' coïncidant sur X sont identiques
- plusieurs DF sur un schéma $(R, \mathcal{A}) \Longrightarrow$ schéma $(R, \mathcal{A}, \mathcal{F})$ où $\mathcal{F}$ est un ensemble de DF

Question

Pour une même relation, plusieurs ensembles de DF possible avec équivalence

Dépendances Fonctionnelles

Définitions

- *Conséquence logique* : si $X \rightarrow Y$ est valide pour toute instance de $(R, \mathcal{A}, \mathcal{F})$
- *Fermeture transitive* : ensemble de DF noté $\mathcal{F}^+$ qui contient toutes les conséquences logiques de $\mathcal{F}$
- *Équivalence* : deux ensembles de DF $\mathcal{F}$ et $\mathcal{F}'$ sont *équivalents* si et seulement si $\mathcal{F}^+ = \mathcal{F}'^+$

Théorème

Soit $\mathcal{F}$ un ensemble de DF, la dépendance $X \rightarrow Y$ est dérivée de $\mathcal{F}$ ($X \rightarrow Y \in \mathcal{F}^+$) si et seulement si $Y \in X^+/\mathcal{F}$.

Fermeture Transitive

Algorithme

Fermeture transitive de X par rapport à $\mathcal{F}$

- **Entrées** : X ensemble d'attibuts de $(R, \mathcal{A}, \mathcal{F})$
- **Sortie** : $X^+/\mathcal{F}$
- **Instructions** :

$X^+ := X$;
$\mathcal{F}' := \mathcal{F}$;
tant que $\exists Y \to Z \in \mathcal{F}' | Y \subseteq X^+$
faire
$\mathcal{F}' := \mathcal{F}' \setminus \{Y \to Z\}$;
$X^+ := X^+ \cup Z$;

Conséquence Logique

Solution

W. W. Armstrong (1974) : système (logique) déductif pour "établir" les conséquences logiques d'un ensemble de DF

Règles

Nom	Conclusion	Hypothèse
Réflexivité	$X \to Y$	si $Y \subseteq X$
Augmentation	$XZ \to YZ$	si $X \to Y$ et $Z \in \mathcal{A}$
Transitivité	$X \to Z$	si $X \to Y$ et $Y \to Z$

Système d'Armstrong

Définitions

Un système déductif est dit :

- *correct* s'il n'engendre que des formules (ici, DF) valables.
- *complet* s'il permet d'obtenir toute formule (ici, DF) valables.
- *adéquat* si et seulement si il est correct et complet.

Théorème

Le système d'Armstrong est adéquat.

Couverture Irredondante Minimale

Objectif : Bon ensemble de DF (éliminer la redondance)

Définitions

- $A \in X$ est un *attribut redondant* dans $X \rightarrow Y$ de $\mathcal{F}$ si et seulement si $Y \in (X \setminus \{A\})^+)$ par rapport à $\mathcal{F}$, (DF minimale à gauche)
- $X \rightarrow Y$ est une *DF redondante* dans $\mathcal{F}$ si et seulement si $(\mathcal{F} \setminus \{X \rightarrow Y\})^+ = \mathcal{F}^+$ ou $Y \in X^+$ par rapport à $\mathcal{F} \setminus \{X \rightarrow Y\}$,
- $\mathcal{F}'$ est une CIM de $\mathcal{F}$ si et seulement si $\mathcal{F}$ et $\mathcal{F}'$ équivalents et les DF de $\mathcal{F}'$ telles que :
  1. sans d'attribut redondant (en partie droite),
  2. pas redondante.

Clef (retour)

Définitions

- Sur-clef K d'un schéma avec DF (R, U, F) ssi $K \rightarrow U$ est une DF dérivable
- Clef K est une sur-clef et minimale (ne contient pas strictement une autre sur-clef) ie. $K \rightarrow \mathcal{A}$ est minimale à gauche

Problème

Calculer les clefs d'un schéma $(R, \mathcal{A}, \mathcal{F})$
Remarque : $\mathcal{A}$ est une sur-clef triviale
$\Longrightarrow$ Un algorithme : minimisation de la DF $\mathcal{A} \rightarrow \mathcal{A}$

Décomposition

Objectif

Casser un schéma $(R, \mathcal{A})$ en plusieurs schémas $(R_i, \mathcal{A}_i)$ sans perte d'information

Définition

Décomposition : les $(R_i, \mathcal{A}_i)$ sont des sous-schémas

Propriétés

- $\mathcal{A} = \bigcup_i \mathcal{A}_i$ (adéquation des attributs)
- $R = \pi_{\mathcal{A}_1}(R) \bowtie \cdots \bowtie \pi_{\mathcal{A}_n}(R)$ (adéquation des N-uplets)

Bonne décomposition

Problème

Décomposition respectant les DF ?

Exemple

Schéma $(R, ABCD, \{AB \to C, C \to D, D \to A\})$
décomposé en deux sous-schémas $(R_1, ABC, \mathcal{F}_1)$ et $(R_2, AD, \mathcal{F}_2)$. Que représentent $\mathcal{F}_1$ et $\mathcal{F}_2$?
A priori, seule $AB \to C$ s'applique sur R_1, mais en fait $C \to A$ est déductible sur R et s'applique aussi sur R_1.

Remarque : complexité exponentielle dans le pire des cas !

Normalisation

Objectif

Normaliser afin de proposer un schéma relationnel "propre" qui soit référentiel et permette d'éviter :

- la redondance des informations,
- les incohérences dans les mises à jour,
- les anomalies lors des insertions/suppressions.

Plusieurs formes normales selon les critères optimisés : 1FN, 2FN, 3FN, FBCN, 4FNF et 5FN.

Formes Normales

Première FN

Toutes les données sont représentées dans des relations binaires (clef, valeur)

Tout est atomique : attribut simple ou monovalué

Toujours le cas dans ce modèle relationnel (pas de relation de second ordre)

Formes Normales

Deuxième FN

$(R, \mathcal{A}, \mathcal{F})$ est en 2FN si et seulement si tout attribut de $\mathcal{A}$ n'appartenant pas à une clef dépend directement d'une clef.

Exemple : Schéma (R,{Appareil, Vol, AéroportDépart, AéroportArrivée},{Vol $\rightarrow$ Appareil,{Vol,AéroportDépart} $\rightarrow$ AéroportArrivée})
Unique clef : K = {Vol, AéroportDépart}
Vol, AéroportDépart $\in$ K. AéroportArrivée dépend directement de K. Mais Appareil ne dépend que de Vol et non de K directement. Ce schéma n'est pas en 2FN.

Formes Normales

Troisième FN

$(R, \mathcal{A}, \mathcal{F})$ avec $\mathcal{F}$ CIM est en 3FN ssi pour toute DF $X \rightarrow A$ de $\mathcal{F}$, on a :

- X est une clef,
- A appartient à une clef.

Forme normale de Boyce-Codd (BCNF)

$(R, \mathcal{A}, \mathcal{F})$ avec $\mathcal{F}$ CIM est en BCNF ssi est en 3FN et pour toute DF $X \rightarrow Y$ de $\mathcal{F}$, X est un clé

Formes Normales

Algorithme (dû à Philip A. Bernstein dans les années 70)

- **Entrée** : un schéma $(R, \mathcal{A}, \mathcal{F})$
- **Sortie** : un ensemble de schémas $\{(R_i, \mathcal{A}_i, \mathcal{F}_i)\}$ en 3FN
- **Instructions** :
  1. $\mathcal{F}' = CIM(F)$;
  2. $\mathcal{K} = Clef(F')$;
  3. Partitionner $\mathcal{F}'$ selon les DF ayant même partie gauche et construire les schémas maximaux pour l'inclusion

$$(R_i, \mathcal{A}_i = Max(X \cup Y), \{X_j \rightarrow Y_j | X_j \cup Y_j \subseteq \mathcal{A}_i\}$$

Formes Normales

- **Instructions** :
  4. Si aucune clef ne figure dans les schémas précédents, ajouter le schéma

$$(R_{i+1}, K, \{K \rightarrow K\})$$

Conclusion

Parfois DF insuffisante

Autres contraintes :

- dépendances multi-valuées (4FN, 5FN)
- dépendances d'inclusion
- dépendances de jointure
- unicité
- etc.

Logiques de description

Introduction

- Représentation des connaissances
- Raisonnement : classification, inférence
- Formalisme logique : (méta-)propriétés
- Langages informatiques : DAML+OIL, Owl

Historique

- Langage de «frames»
- Réseaux sémantiques
- Graphes conceptuels
- Systèmes à base de connaissances : KL-ONE, BACK et LOOM
- Web sémantique et ontologie

dans le contexte de la logique formelle : propositionnelle et prédicative (longue histoire)

Syntaxe

La syntaxe des langages de description est donnée par 3 types : les individus I, les concepts C et les rôles R.
Il y a 2 niveaux d'assertions ou «boîtes» (box) :

- le général qui portent essentiellement sur les concepts et les rôles,
- le factuel qui portent essentiellement sur les individus.

Individus

Ce sont essentiellement des identifiants, donc $\mathcal{I}$ est la donnée d'un ensemble d'atome.

$I \equiv \mathcal{I} \mid$ atome

Exemple : Anne, Sophie, Robert, David

Rôles

$R \equiv$	$\mathcal{R}$		atome
	$\top_R$	anything	
	$\bot_R$	nothing	
	$R \sqcap R$	r-and	et
	R/C	range	restreint à
	R^{-1}	inverse	inverse de
	$R \circ R$	compose	composée de

Exemple : estParentDe

Concepts

$C \equiv$	$\mathcal{C}$		atome
	$\top$	anything	
	$\bot$	nothing	
	$\neg\mathcal{C}$	a-not	non atomique
	$\neg C$	not	non généralisée
	$C \sqcap C$	c-and	et
	$C \sqcup C$	c-or	ou
	$\forall R.C$	all	tous
	$\exists R$	some	quelque
	$\geq nR$	atleast	au moins
	$\leq nR$	atmost	au plus
	$\exists R.C$	c-some	
	$\geq nR.C$	c-atleast	au moins
	$\leq nR.C$	c-atmost	au plus
	$\{\mathcal{I}, \ldots, \mathcal{I}\}$	one-of	un parmis

Exemple : Male, Femelle, Humain, Homme, Femme, Pere, Mere

Propriétés

Quelques équivalences :

$$\begin{aligned} \exists R.\top &\equiv \exists R \\ \leq nR.\top &\equiv \leq R \\ \geq nR.\top &\equiv \geq R \end{aligned}$$

Terminologie $\mathcal{T}$

La $\mathcal{T} - Box$ est la boîte dite «terminologique». Elle détermine la structure conceptuelle à l'aide d'un ensemble de deux formes d'assertion :

- la *subsomption* $C \sqsubseteq C$,
- la *définition* $C \doteq C$, qui est en fait une double subsomption (symétrique).

La subsomption représente le fait que tous les individus d'un concept sont aussi des individus d'un autre concept. Autrement dit, son interprétation correspond à l'inclusion. La définition représente le fait que tous les individus d'un concept sont exactement les individus d'un autre concept. Généralement l'un des deux membres de l'égalité est plus complexe que l'autre. Autrement dit, son interprétation correspond à l'égalité.

Assertion $\mathcal{A}$

La $\mathcal{A} - Box$ est la boîte dite «ontologique». Elle détermine la structure individuelle à l'aide de deux formes d'assertion :

- la *conception* $C(I)$,
- la *relation* $R(I, I)$.

Exemple

$\mathcal{T} - Box$	$\mathcal{A} - Box$
Femelle $\sqsubseteq \top \sqcap \neg$ Male	
Male $\sqsubseteq \top \sqcap \neg$ Femelle	Humain(Anne)
Animal $\doteq$ Male $\sqcup$ Femelle	Femelle(Anne)
Humain $\sqsubseteq$ Animal	Femme(Sophie)
Femme $\doteq$ Humain $\sqcap$ Femelle	Humain(Robert)
Homme $\doteq$ Humain $\sqcap \neg$ Femelle	$\neg$ Femelle(Robert)
Mere $\doteq$ Femme $\sqcap \exists$ estParentDe	Homme(David)
Pere $\doteq$ Homme $\sqcap \exists$ estParentDe	estParentDe(Sophie,Anne)
MereSansFille $\doteq$ Mere $\sqcap \forall$ estParentDe . $\neg$ Femme	estParentDe(Robert,David)
estParentDe $\sqsubseteq \top_R$	

Interprétation

Une interprétation est la donnée d'un domaine $\mathcal{D}$ et d'une fonction $\|-\| : \mathcal{L} \longrightarrow \mathcal{D}$, devant être définie pour chacun des trois types et de manière à respecter les assertions.

Les individus

L'interprétation de chaque individu est un élément du domaine, autrement dit $\|I\| \subseteq \mathcal{D}$. Les contextes complètent l'interprétation. Pour cela, on donne l'interprétation de chaque individu de la $\mathcal{A} - Box$.

Les concepts

L'interprétation d'un concept est une sous-ensemble du domaine.

$$
\begin{aligned}
\|\top\| &= \mathcal{D} \\
\|\bot\| &= \emptyset \\
\|\neg C\| &= \mathcal{D} \setminus \|C\| \\
\|C \sqcap C\| &= \|C\| \cap \|C\| \\
\|C \sqcup C\| &= \|C\| \cup \|C\| \\
\|\forall R.C\| &= \{d \in \mathcal{D} \mid \forall e : (d,e) \in \|R\| \Rightarrow e \in \|C\|\} \\
\|\exists R\| &= \{d \in \mathcal{D} \mid \exists e : (d,e) \in \|R\|\} \\
\|\geq nR\| &= \{d \in \mathcal{D} \mid \sharp\{e \mid (d,e) \in \|R\|\} \geq n\} \\
\|\leq nR\| &= \{d \in \mathcal{D} \mid \sharp\{e \mid (d,e) \in \|R\|\} \leq n\}
\end{aligned}
$$

Les rôles

L'interprétation d'un rôle est un sous-ensemble du produit cartésien sur le domaine.

$$
\begin{aligned}
\|R \sqcap R\| &= \|R\| \cap \|R\| \\
\|R/C\| &= \{\} \\
\|R^{-1}\| &= \{(e,d) \mid (d,e) \in \|R\|\} \\
\|R \circ R\| &= \{(d,f) \mid (d,e) \in \|R\| \cup (e,f) \in \|R\|\}
\end{aligned}
$$

Subsomption

La subsomption est une contrainte forte car elle doit être respecter toute interprétation $\|-\|$.

Inférence

0.1 Au niveau terminologique

Étant donnée une terminologie, les principaux problèmes d'inférence sont :

- satisfaction : un concept C est satisfaisable si et seulement si il existe un modèle $\|-\|$ tel que $\|C\| \neq \emptyset$
- subsomption : un concept C_1 est subsumé par un concept C_2 si et seulement si $\|C_1\| \subset \|C_2\|$ pour tout modèle $\|-\|$,
- équivalence : un concept C_1 est équivalent à un concept C_2 si et seulement si $\|C_1\| = \|C_2\|$ pour tout modèle$\|-\|$,
- disjonction : deux concepts C_1 et C_2 sont disjoints si et seulement si $\|C_1\| \backslash \|C_2\| \emptyset$ pour tout modèle $\|-\|$.

Inférence (2)

0.2 Au niveau factuel

Étant données une terminologie $\mathcal{T}$ et une ontologie $\mathcal{A}$:

- cohérence : $\mathcal{A}$ est cohérente par rapport à $\mathcal{T}$ si et seulement si il existe un modèle,
- vérification d'instance : vérifier par inférence si une assertion $C(I)$ est vraie pour tout modèle,
- vérification de rôle : vérifier par inférence si une assertion $R(I, I)$ est vraie pour tout modèle,
- extraction : inférer les individus d'un concept pour tout modèle.

Algèbre d'intervalles

Introduction

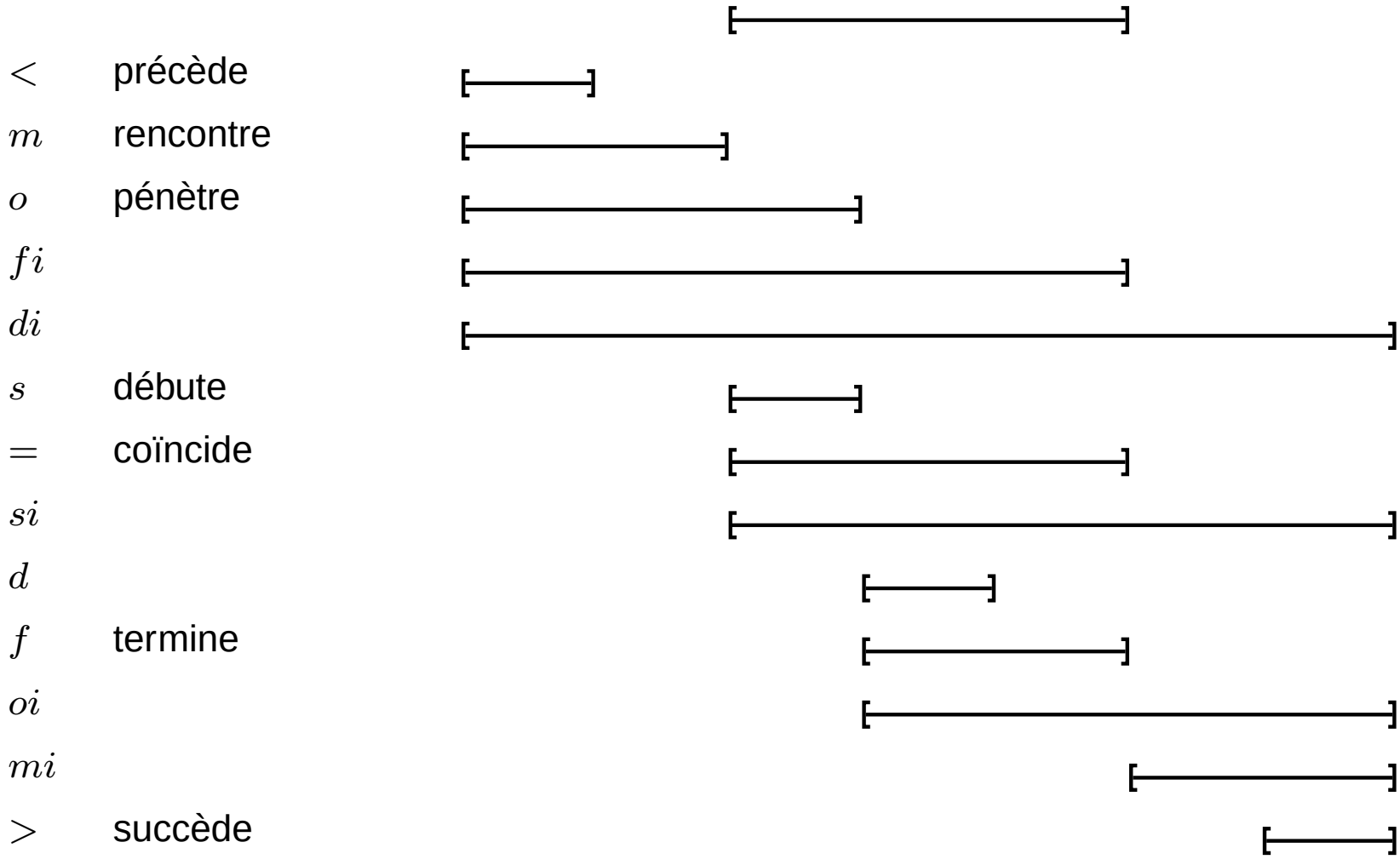
Situation ensembliste, deux ensembles E et F :

- $E = F$,
- $E \subset F$,
- $E \cap F = \emptyset$,
- autres ?

Définitions

Étant donné un ensemble E muni d'une relation d'ordre total $<$, on appelle *intervalle* un sous-ensemble de E. Un tel objet est isomorphe à un élément du produit cartésien $E \times E$. On note A^- (resp. A^+) la *borne inférieure* (resp. *supérieure*) de A.

Relations d'intervalles



$$\mathcal{RI} \leftarrow \{<, m, o, fi, di, s, =, si, d, f, oi, mi, >\}$$

Propriétés

Inverse :

- $A < B \Longleftrightarrow B > A$
- $A\ m\ B \Longleftrightarrow B\ mi\ A$
- $A\ o\ B \Longleftrightarrow B\ oi\ A$
- $A\ fi\ B \Longleftrightarrow B\ f\ A$
- $A\ di\ B \Longleftrightarrow B\ d\ A$
- $A\ s\ B \Longleftrightarrow B\ si\ A$

Soit donc 7 relations.
On retrouve : $A \subset B \Leftrightarrow A\ d\ B \vee A\ s\ B \vee A\ f\ B$.

Relations aux bornes

- Par définition :

A	$<$	m	o	fi	di	s	$=$	si	d	f	oi	mi	$>$	B
A^-	$<$	$<$	$<$	$<$	$<$	$=$	$=$	$=$	$>$	$>$	$>$	$\star$	$>$	B^-
A^-	$<$	$<$	$<$	$<$	$<$	$<$	$\star$	$\star$	$<$	$\star$	$<$	$=$	$>$	B^+
A^+	$<$	$=$	$>$	$\star$	$>$	$\star$	$\star$	$>$	$>$	$>$	$>$	$>$	$>$	B^-
A^+	$<$	$\star$	$<$	$=$	$>$	$<$	$=$	$>$	$<$	$=$	$>$	$>$	$>$	B^+

- Quel choix pour les cases $\star$?

Cas particulier : A ou B points

Exemple : $B^- = B^+$

A	$<$	m	o	fi	di	s	$=$	si	d	f	oi	mi	$>$	B
A^-	$<$	$<$		$<$	$<$			$=$				$=$	$>$	$B^- = B^+$
A^+	$<$	$=$		$=$	$>$			$>$				$>$	$>$	$B^- = B^+$

en composant (union) les lignes respectives.
Mais il y a alors des ambiguïtés dues aux équivalences :

- $A\ fi\ B \Longleftrightarrow A\ m\ B$,
- $A\ si\ B \Longleftrightarrow A\ mi\ B$.

Soit donc 5 relations possibles.

Relations aux bornes (bilan)

A	$<$	m	o	fi	di	s	$=$	si	d	f	oi	mi	$>$	B
A^-	$<$	$<$	$<$	$<$	$<$	$=$	$=$	$=$	$>$	$>$	$>$	$>$	$>$	B^-
A^-	$<$	$<$	$<$	$<$	$<$	$<$	$\leq$	$\leq$	$<$	$\leq$	$<$	$=$	$>$	B^+
A^+	$<$	$=$	$>$	$\geq$	$>$	$\geq$	$\geq$	$>$	$>$	$>$	$>$	$>$	$>$	B^-
A^+	$<$	$<$	$<$	$=$	$>$	$<$	$=$	$>$	$<$	$=$	$>$	$>$	$>$	B^+

Variable

On peut généraliser à un sous-ensemble des relations $\mathcal{RI}$, dit *alternative* et noté R_{AB}, pour décrire les relations possibles entre deux intervalles A et B. On parle de *déterminée* si cet ensemble est réduit à un seul élément, et on peut alors noter indifféremment $A\ \{?\}\ B$ et $A\ ?\ B$, autrement (plus d'un élément) on parle d'*indéterminée*.
Ainsi, $A\ \{<, m\}\ B$ signifie $A\ \{<\}\ B$ ou $A\ \{m\}\ B$.
Bien sûr, un problème classique consiste à pouvoir alors déterminer toutes les possibilités entre des intervalles étant données des certitudes entre eux.
Par exemple, en considérant $A\ \{d\}\ B$, $B\ \{<, m\}\ C$, et $D\ \{d\}\ C$, on peut montrer que $A\ \{<\}\ D$.
On a : $\subset \equiv \{d, s, f\}$.

Transitivité

Étant donnés trois intervalles A, B et C, le tableau T ci-après donne les 13×13 alternatives de A et C issues des certitudes de A et B, et de B et C.

	$<$	m	o	fi	di	s
$<$	$<$	$<$	$<$	$<$	$<$	$<$
m	$<$	$<$	$<$	$<$	$<$	m
o	$<$	$<$	$<, o, m$	$<, o, m$	$<, di, o, m, fi$	o
fi	$<$	m	o	fi	di	o
di	$<, di, o, m, fi$	di, o, fi	di, o, fi	di	di	di, o, fi
s	$<$	$<$	$<, o, m$	$<, o, m$	$<, di, o, m, fi$	s
$=$	$<$	m	o	fi	di	s
si	$<, di, o, m, fi$	di, o, fi	di, o, fi	di	di	$=, s, si$
d	$<$	$<$	$<, d, o, m, s$	$<, d, o, m, s$	$\mathcal{RI}$	d
f	$<$	m	d, o, s	$=, f, fi$	$>, di, oi, mi, si$	d
oi	$<, di, o, m, fi$	di, o, fi	$=, o, oi, s, si, f, fi$	di, oi, si	$>, di, oi, mi, si$	d, oi, f
mi	$<, di, o, m, fi$	$=, s, si$	d, oi, f	mi	$>$	d, oi, f
$>$	$\mathcal{RI}$	$>, d, oi, mi, f$	$>, d, oi, mi, f$	$>$	$>$	$>, d, oi, mi, f$

Fermeture

Étant données deux variables R_{AB} et R_{BC}, on peut calculer la variable entre A et C, R_{AC}, par l'algorithme suivant :

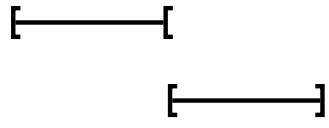
- entrée : R_{AB} et R_{BC}
- sortie : R_{AC}
- instruction :
  $R_{AC} \leftarrow \emptyset$;
  $\forall r \in R_{AB}$:
    $\forall s \in R_{BC}$:
      $R_{AC} \leftarrow R_{AC} \cup T(r, s)$;

Étant donnés un ensemble d'intervalles et des relations possibles entre eux, il est possible de calculer l'ensemble de toutes les alternatives possibles.

Intervalles dans les séquences

Intervalles = segments sur une séquence biologique, soit des éléments $\mathbb{N}^2$.

- retrouver les relations de l'ontologie IMGT®
- cependant considérer des relations comme :
  $[1;9] \star [10;15]$
- solution ?



$[1;9] \mapsto [1;10[$

Exemple des V,D,J-GENE