

## Représentation des Connaissances : Réseaux Sémantiques, Frames & Scripts



Bernard ESPINASSE  
Professeur à l'Université d'Aix-Marseille



2004

- Introduction : grandes méthodes de la représentation des connaissances
- Les Réseaux Sémantiques
  - Définition, concepts, héritage, partition, quantification (Hendrix)
  - Logique & Réseaux sémantiques (Schubert & Cerone)
  - Interprétation des connaissances et niveaux de compréhension
  - Intérêts et limites des réseaux sémantiques
- Les Frames et Scripts : l'approche objet en IA
  - Notion de Frame (Schéma) : composant, attribut, aspect
  - Héritage et inférences dans les frames
  - Introduction aux Scripts

## Introduction : grandes méthodes de la RdC

### Logique formelle :

- logiques standards : logique des propositions, logique des prédicats
- logiques non-standard : logique modales, logiques temporelles, déontique (de la préférence), multivalentes, intuitionniste, floues, non-monotone

### Modèles psychologiques :

- la sémantique psychologique
- les prototypes
- les dépendances conceptuelles et leurs dérivées

### Modèles informatiques :

- bases de données (modèle relationnel)
- les réseaux sémantiques \*
- les représentations orientées objets (frames, smalltalk,...) \*
- les règles de production (systèmes experts : Mycin, Snark, OPS, CLIPS, ...)

## Les réseaux sémantiques : définition

- conçus à l'origine en linguistique pour devenir plus tard un langage pour la représentation de concepts très divers, une structure informatique utilisée en IA (QUILLIAN / COLLINS 1966)
- un réseau sémantique est un graphe composé de:
  - un ensemble de **noeuds** étiquetés : *représentant généralement des objets*,
  - un ensemble de **liens** orientés et étiquetés entre ces noeuds : *représentant généralement des relations entre des objets*,
  - un ensemble d'**opérations** d'exploitation de ce graphe, *constituants les mécanismes de raisonnement*
- **représentation graphique:**

facilite la lecture, ne correspond généralement pas au formalisme d'implémentation,



- **représentation non-graphique:**

(SENTIR, GASTON, CHLOE)

## Les réseaux sémantiques : concepts

### les NOEUDES

- atomiques : entités élémentaires (valeurs, individus,...)
- complexes : entités complexes (propositions, phrases,...)
- ils doivent être typés : concept, individu, action, proposition, etc...

### les LIENS

- structuraux : indépendants de la sémantique du domaine,
- spécifiques : dépendants de la sémantique du domaine,

*il faut essayer d'augmenter la proportion des liens structuraux par rapport aux liens spécifiques*

### les OPERATIONS

- souvent représentées par le programme,
- doivent être définies clairement,

## Les réseaux sémantiques : noeuds concepts

"les canaris / sont des /oiseaux"

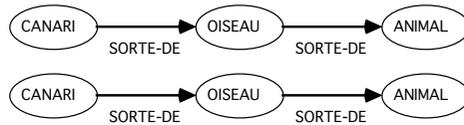
canaris et oiseaux =

**concepts** (nom communs) --> classe

sont des =

**relation** --> inclusion de classes

### liens sorte-de



- lien **structurel** indépendant du domaine
- représente une **inclusion**
  - de **propriétés** (pt de vue intentionnel, cas général)
  - d'**individus** (pt de vue extensionnel)

## Les réseaux sémantiques : noeuds individus

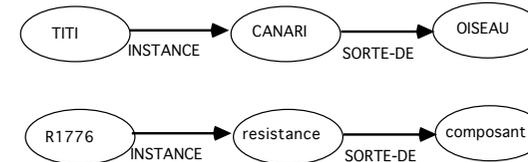
"Titi / est un / canari"

canari = **concepts**

Titi = **individu** (nom propre) --> élément d'un ensemble

est un = **relation** --> appartenance d'un élément à une classe

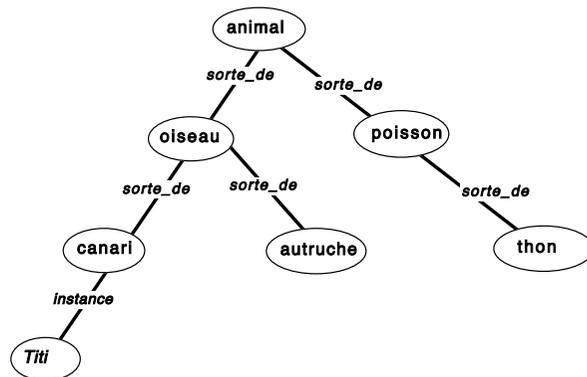
### lien d'instance



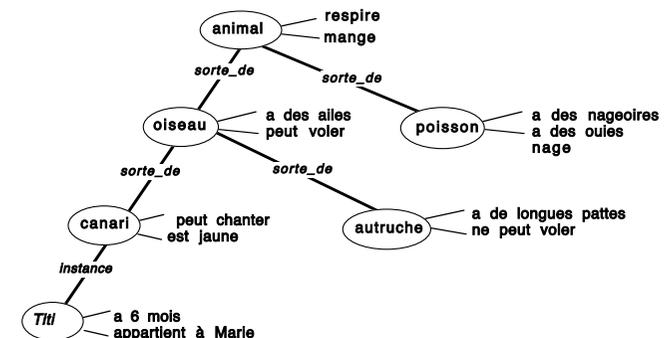
- lien **structurel**

## Les réseaux sémantiques : taxonomies

- liens **sorte-de** et noeuds **concepts** + liens **d'instance** et noeuds **individu** :



## Les réseaux sémantiques : propriétés

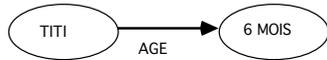


- des propriétés "non-analysées" sont affectées à chaque noeud
- simple
- ne permet pas de répondre à des questions comme :  
 "quel est l'âge de Titi ?"      "quelle est la couleur des canaris ?"  
 -> **notion d'attributs**

## Les réseaux sémantiques : attributs (1)

- **attribut** = relation qui relie un noeud concept ou un noeud individu à une valeur ou propriété

"l'age de Titi est de 6 mois"

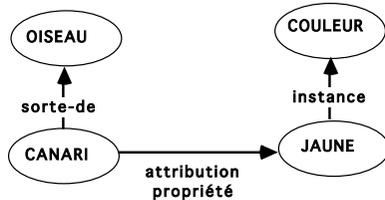


"la couleur des canaris est le jaune"



- **lien spécifique** dont le sens dépend du domaine d'application -> interprétation ad-hoc,

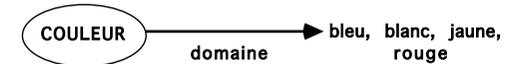
- on peut le rendre plus structural en créant un **noeud-attribut**:



## Les réseaux sémantiques : attributs (2)

- peut être vu comme une classe sémantique dont les instances sont des propriétés

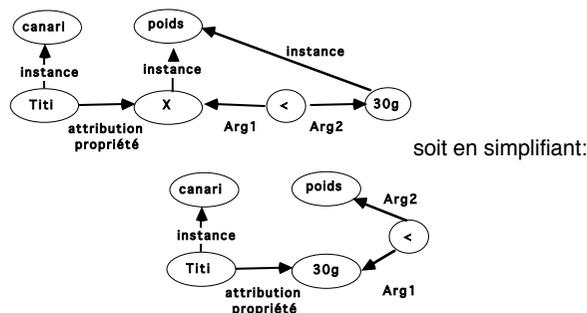
- un attribut peut lui-même être caractérisé :



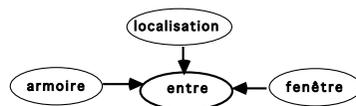
- **domaine** = relation structurelle permettant de vérifier des contraintes d'intégrités.

## Rapports attribut / valeur

- un noeud-attribut peut être relié à une ou plusieurs valeurs par l'intermédiaire d'un opérateur:



- cet opérateur peut être n-aire:



## Le modèle de M.R.QUILLIAN

- chaque concept nœud est considéré comme un "plan" (plane) qui contenait sa définition.
- des pointeurs à l'intérieur d'un plan définissent la structure de la définition du plan.
- dans ce plan peuvent être définies:
  - des sous-classes (),
  - des modifications de définition (),
  - des disjonction (ou), des disjonctions (et),
  - des relations syntaxiques du type sujet /objet (ex: "les hommes utilise cette plante...": *utilise* (sujet: hommes; objet: A)).
- un pointeur partant d'un mot (nœud) d'un plan donné pointe vers un autre plan définissant le mot.
- un mot utilisé dans la définition d'un plan puisse être à son tour défini dans un autre plan externe conduit à la distinction fondamentale de type et d'occurrence; *chaque mot, nœud d'un plan est défini dans un plan unique, la tête de ce plan étant le nœud type.*
- **techniques d'inférences:** *étant donné 2 mots, les possibles relations entre eux pouvaient être inférées par une recherche heuristique en largeur d'abord dans la zone avoisinante des mots en question. Cette recherche était assurée par propagation à travers le réseau.*

## Héritage de propriétés

L'étude des temps de réaction de sujets humains semblent confirmer l'existence probable d'un modèle hiérarchique de la mémoire humaine.

COLLINS et QUILLIAN définissent alors des Réseaux Sémantiques comme :

- des sur-ensembles de hiérarchies de concepts

*Ex : animal, oiseau, canari, dans lesquels chaque nœud est un concept auquel est rattaché un ensemble de propriétés, par exemple: chante, est jaune,....*

- des propriétés plus générales sont rattachées sur la hiérarchie de généralisation

*Ex : elles permettent d'affirmer "un canari à une peau" et "un canari a des ailes";*

→ ainsi sont rajoutées :

- au concept « animal » la propriété « à une peau »
- au concept « oiseau » la propriété « à des ailes »

→ notion fondamentale d'héritage de propriétés dans un réseau sémantique

## Notion d'héritage

- liens de type « est\_un » ou « sorte\_de » reliant un concept à un autre concept plus élevé :

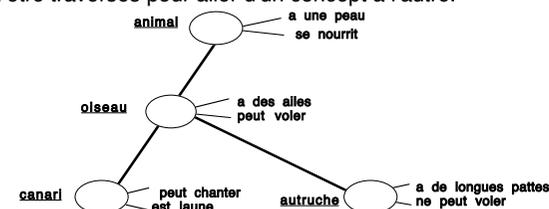
*par exemple: "canari" est une sorte de "oiseau"*

- héritage des propriétés rattachées au concept père au concept fils:

*Ainsi, on pourra dire que « le canari a des ailes et une peau » en remontant les liens « sorte\_de »*

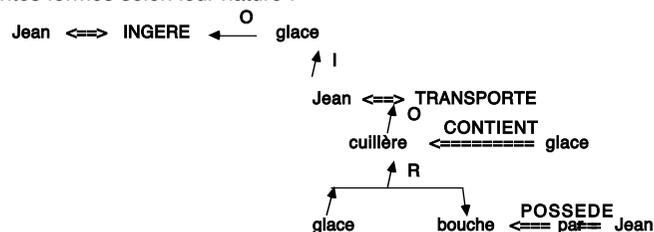
- le principe d'héritage permet :

- de nombreuses déductions automatiques
- de définir la notion de **distance sémantique** entre 2 concepts = nombre de liens devant être traversés pour aller d'un concept à l'autre.



## Dépendance conceptuelle (1)

- notion de **dépendance conceptuelle** introduite par R.C.SCHANK (linguistique, 1970)
- Une conceptualisation se compose
  - d'une action primitive
  - d'un certain nombre de liens entre des objets qui interviennent dans cette action, comme un "instrument" et une "direction".
- un objet peut être un objet physique (Jean, un livre,...) ou une autre conceptualisation.
- dans les diagrammes de dépendances conceptuelles, les liens, représentés par des flèches ont différentes formes selon leur nature :



## Dépendance conceptuelle (2)

- <==> lie une action à son acteur, dans l'exemple: Jean **INGERE**, Jean **TRANSPORTE**
  - --> lie une action à un objet qui subit l'action:
    - le lien I indique l'instrument de la conceptualisation, ici l'instrument d'une action est lui même une conceptualisation.
    - les liens O indiquent l'Objet sur lequel porte l'action
  - le lien à trois flèches est la **structure direction** qui est utilisé pour certaines actions où l'objet qui subit l'action subit un déplacement, dans l'exemple:
    - le lien R, indique le lien Récipient, *glace est le point de départ et bouche est le point d'arrivé.*
- Les verbes **INGERE**, **TRANSPORTE** = actions primitives auxquels sont associés des structures de liens particulières.

Des verbes de plus haut niveau doivent être décomposés en structures canoniques d'actions primitives comme celles-ci.

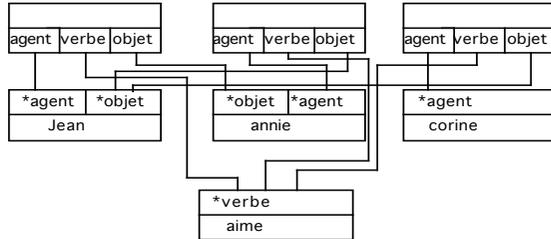
→ SCHANK spécifie un ensemble de connaissances primitives sur lesquelles des concepts peuvent être construits.

- RUMELHART, NORMAN: distinction de plusieurs types de nœuds dans un réseau sémantique: nœuds liés à des concepts, des événements, des épisodes (séquences d'événements agrégés ensembles).

## Réseaux sémantiques et structures de données

[SHAPINO, CERONE, SCHUBERT, LEVESQUE et MYLOPOULOS]

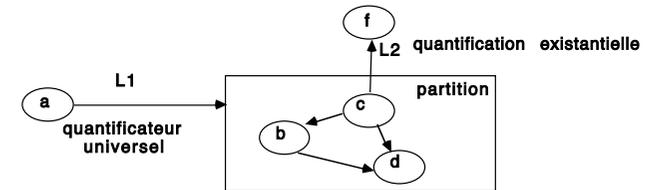
- 2 niveaux pour SHAPINO : niveau de "l'item" (conceptuel) et niveau du "système":
  - relations systèmes : liens étiquetés et sémantique assurée par des routines spécifiques les manipulant,
  - relations items : considérés comme des concepts, représentés par des nœuds (items):



- séparation explicite des liens primitifs sous-jacents des relations conceptuelles
- explique comment des règles de déduction peuvent être encodées directement sur ce formalisme.

## Partition dans les réseaux sémantiques

- **partition** [Hendrix] = regroupement de nœuds et d'arcs du réseau dans des espaces spécifiant la portée de relations
- intérêts des partitions :
  - définition de contextes
  - permet la quantification :

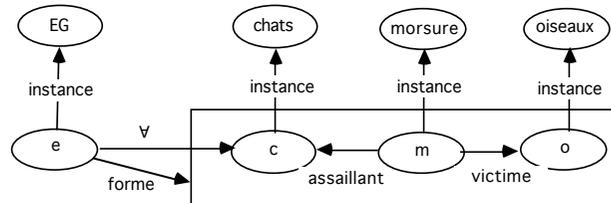


- cadres définissent l'étendue des identificateurs universels
- lien L1 = quantification universelle, quelque soit a, pointe sur un cadre représentant l'étendue de la variable quantifiée universellement.
- lien L2 = quantificateur existentiel explicite sur le nœud f par rapport au nœud c

- certains systèmes experts (Prospector) l'utilise

## Réseaux sémantiques et quantification (Hendrix)

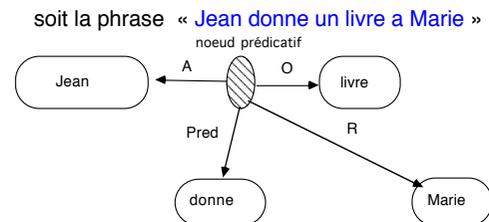
- quantification traitée par la notion de partition
- Soit le fait à représenter suivant : « tout chat a mordu un oiseau »
  - représentation logique :  $\forall x \text{ chat}(x) \rightarrow (\exists y \text{ oiseau}(y) \wedge \text{mordre}(x,y))$
  - encodage de la variable quantifiée universellement x en utilisant une partition (cadre rectangulaire) :



- les nœuds c, m, o sont des instances de chats, morsure, oiseaux,
- le cadre introduit dans le réseau définit l'étendue de l'identificateur universel,
- le nœud e représente l'assertion à représenter, instance de l'ensemble des énoncés généraux EG sur le monde,
- chaque élément de EG possède :
  - une connexion « forme » pointant vers le cadre de la partition et énonce l'affirmation,
  - une ou plusieurs connexions « V » pointant vers chaque variable quantifiée universellement, ici variable c (les variables m et o sont ici quantifiées existentiellement).

## Logique & Réseaux sémantiques (Schubert & Cerone)

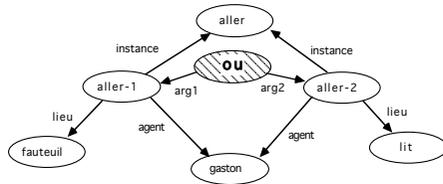
- introduction de la logique des prédicats du premier ordre
- introduction d'un "nœud prédicatif" instancié en lui associant :
  - un pointeur vers le prédicat
  - un pointeur vers chaque argument du prédicat
- Ex :



soit : **donne (A, O, R)**  
avec : A (agent) = Jean; O (objet) = livre; R (receveur) = Marie

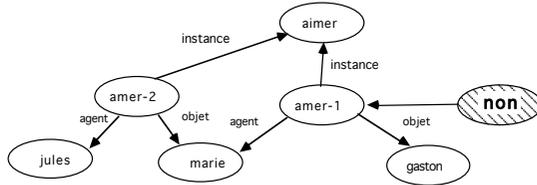
## Connecteurs logiques ET,OU

• Soit la phrase suivante: « **Gaston ira sur le lit ou sur le fauteuil** », une représentation :



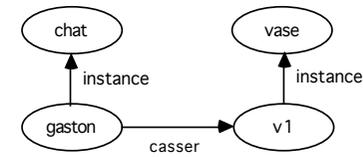
## Représentation de la négation

• Soit la phrase: « **Marie n'aime pas Gaston** », une représentation :

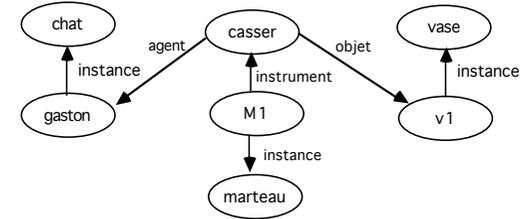


## Représentation d'événements ou d'actions

Représenter l'événement : « **Gaston casse le vase** »:



- le lien « **casser** » et **spécifique**. On peut s'en séparer en le traduisant par des liens plus structurels : agent, objet, instrument, temps, lieu, .... :



## Interprétation des connaissances dans les RS

- l'accès aux données stockées dans un RS n'est **pas assuré par le réseau lui-même**
- appel à un **interpréteur** (transformer les données du réseau en connaissances opératoires)

• on doit disposer alors de:

- soit d'un **langage élaboré de navigation et d'inférence** dans le réseau,
- soit d'un **langage limité à l'accès** dans le réseau + autre **programme assurant les inférences**

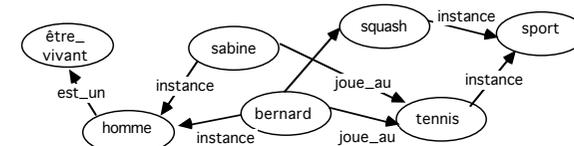
**Exemple dans SNARK = interpréteur de Réseau sémantiques :**

- moteur d'inférences à règles de production à variables
- logique d'ordre 0,1,2
- faits = triplets (O1, R, O2)

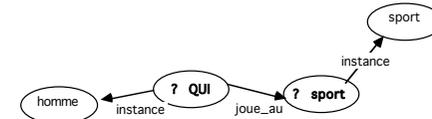


## Interprétation des connaissances dans les RS

On dispose d'une base de connaissances organisée en réseau sémantique :



• la question « **quelqu'un fait-il du sport ?** » représentée par le fragment de réseau suivant :



• les réponses seront :

**sabine joue\_au tennis**  
**bernard joue\_au tennis**  
**bernard joue\_au squash**

→ si réseau important → problèmes combinatoires ....

## Niveaux de compréhension d'un RS

- la caractéristique commune des RS est seulement leur **aspect associatif** (éléments de base de la représentation différant d'un projet à l'autre)

→ définition de **5 niveaux de compréhension d'usage** des réseaux sémantiques [R.J. Brachman]:

- **linguistique**
- **conceptuel**
- **épistémologique**
- **logique**
- **d'implémentation**

- ces niveaux permettent de décrire les différentes sorte d'utilisation des réseaux sémantiques.

## Intérêts des réseaux sémantiques (1)

**Axes organisationnels qu'ils offrent pour structurer une base de connaissances :**

### • La classification, ensembles/sur-ensembles

Un objet peut être associé avec son ou ses types génériques,

*Ex : Titi peut être associé à oiseau, animal.*

- conduit à la distinction fondamentale de type (canari) et d'occurrence (Titi).
- peut être récurive -> définir des méta-types ayant pour instance d'autres types.

### • Agrégation

- rattacher à un objet des propriétés ou d'autres objets y intervenant comme parties.

*Ex : Titi, vu comme objet physique possède des ailes, une tête et une queue, considéré dans son environnement, il possède un nid, un territoire, un chant, une nourriture.*

- peut être appliqué récurivement: un composant peut être à son tour composé d'autres composants

## Intérêts des réseaux sémantiques (2)

### • La généralisation, la spécialisation

relie un type à un autre type plus générique,

*Ex : oiseau à animal*

- La généralisation (lien « sorte\_de ») = un ordre partiel organisant 2 types dans une généralisation ou une hiérarchie.
- économie de place en mémoire (propriétés associées à des types généraux hérités par d'autres types plus spécialisés).
- généralisation plus facile de grandes bases de connaissances (bases de données)

### • La partition

regroupe des objets et éléments de relations dans des partitions qui sont organisées de façon hiérarchiques;

*Ex : si une partition P1 est au-dessous d'une autre P2, toute chose visible ou présente dans P2 l'est aussi dans P1, sans pour autant l'y avoir été spécifiée.*

- principal intérêt = **quantification** [Hendrix], la représentation du **temps** et de **l'hypothétique**. [Cohen 78].

## Conclusion sur les Réseaux Sémantiques

### Forces des RS :

- les objectifs d'extraction de connaissance dans la base de connaissances s'expriment en **chemins de traversée sur la structure même de la base**
- possèdent des **principes d'organisation relativement puissants** (généralisation, partition, agrégation) permettant de **structurer** la base de connaissances
- **formalisme graphique** : bonne **compréhension**, intéressant à un **premier stade de formalisation de la connaissance**
- formalisation **déclarative** : finesse et cohérence de représentation des concepts

### Faiblesses des RS :

- **manque de sémantique formelle** et de **terminologie standard**
- **interprétation difficile** des connaissances : toujours un compromis à faire entre *complexité d'une structure de données* et *complexité de l'interpréteur*
- **critique** si taille du réseau importante (nb de nœuds et liens) → **explosion combinatoire**

→ **de nouveaux outils de représentation ont été développés : les représentations objet (Frame & Scripts)**

## Les Frames, scripts, objets ...

- **Faiblesses des Réseaux Sémantiques** (représentation/interprétation des connaissances)

→ de nouveaux outils ont été développés : **représentations « orientées objet »** :

### Frames, Scripts, langages objets

- représentations des connaissances **MIXTE** :

#### **déclaratives et procédurales**

afin de :

- mieux représenter des **connaissances par nature structurées**
- d'en **faciliter une part d'interprétation**

## L'approche objet en IA

- **objet 1 — relation → objet 2**

• **boite — est\_sur → table**

• **boite — est\_de\_couleur → rouge**

on peut regarder ces descriptions de 2 façons différentes :

- **en privilégiant les relations** :

→ passer par la relation pour accéder aux objets qu'elle relie :

**est\_sur (boite, table)**

*réseaux sémantiques, logique, ...*

- **en privilégiant les objets** :

→ accéder à un objet pour vérifier s'il possède (ou non) une relation avec un autre objet

• **boite** : **est\_sur** : **table**

• **table** : **est\_sous** : **boite**

*frames, objets, ...*

## L'approche objet en IA

- de façon générale, en informatique traditionnelle :

- les programmes = **éléments actifs**

→ *exécutent les opérations sur des ensembles de données*

- les données = **éléments passifs**

→ *subissent les opérations des programmes*

*d'où : programmer = définir les opérations qui agissent sur de données*

- avec les langages orientés objets : **renversement !!!!**

- les objets (données) = **éléments actifs**

- les objets sont caractérisés par les **opérations** qu'ils connaissent

## Notion de Frame (Schéma)

- Minsky (1975)

- **structure de données générale**

- permet de **décrire la connaissance** que l'on a sur des **objets** (concrets ou abstraits)

- on peut rattacher 2 types d'informations à un Frame :

→ **des informations relatives à la description des objets : partie déclarative**

- propriétés
- relations d'états
- relations entre les objets, ...

→ **des informations relatives à la manipulation des objets : partie procédurale**

c.a.d. les opérations (procédures & méthodes) connues par l'objet :

- définition du contexte d'activation de l'objet
- comment calculer une propriété
- action à exécuter dans un contexte donné, ...

## Notion de Frame

### • 2 types de Frames :

#### • les frames « prototypes » :

- représentent en **intention** une **classe d'objets**
- description du **contexte** attaché à cette classe

#### • les frames « instances » :

- réalisations **particulières** d'une classe donnée
- description des **individus** d'une classe

#### Ex:

- oiseau → classe : **prototype**
- canari → classe : **prototype**
- titi → individu : **instance**

## Composants d'un Frame : attributs et aspects (1)

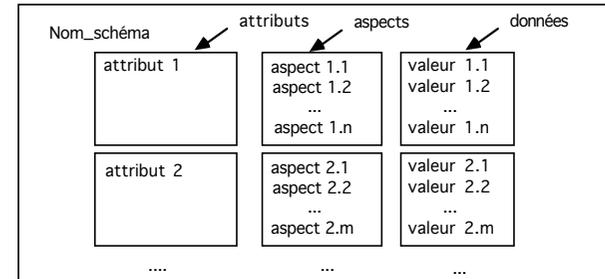
### • attribut (slot) = information d'un frame permettant d'introduire :

- des **propriétés** (1 ou n) **décrivant le frame** :
  - dans frame prototype : introduction de **valeurs permises**
  - dans frame instance : introduction de **valeurs effectives**

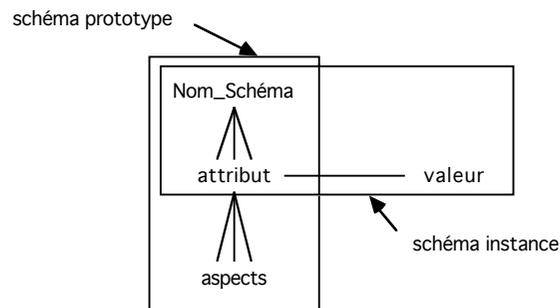
### • des **relations** ( un-aire ou n-aire) **entre frames**

### • un attribut est **structuré** par des **aspects**

### • **aspect** = chaque aspect introduit une valeur élément de description de l'attribut



## Composants d'un Frame : attributs et aspects (2)



- chaque attribut d'une instance doit avoir l'**aspect "valeur"**,
- le frame d'instance ne diffère du frame prototype correspondant que par la donnée supplémentaire de la valeur de l'attribut,
- une instance **hérite** directement du frame de sa classe
- une instance peut être **partielle** ou **complète**,
- une instance de frame est liée à son frame prototype par l'attribut "**est\_un**"
- 2 frames prototypes sont liés par l'attribut "**sorte\_de**"

## Attribut d'un Frame

### • un attribut peut :

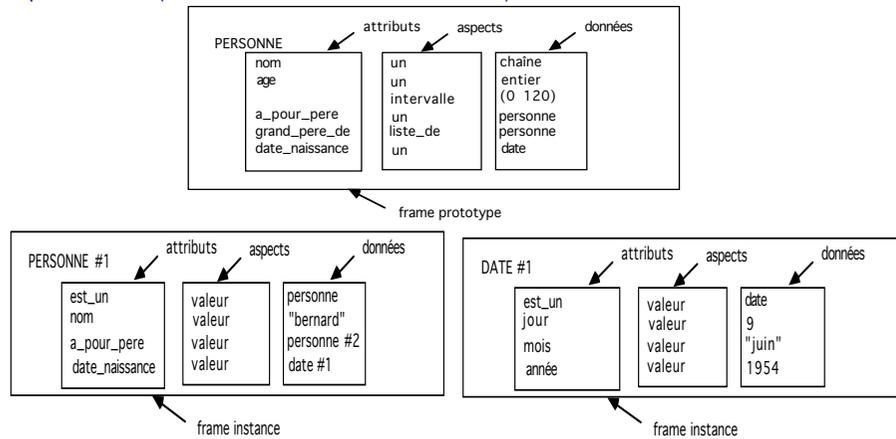
- permettre d'**introduire des valeurs effectives** (frame instance),
- être un **pointeur sur un frame plus général**, plus **spécifique** voire alternatif (vers frame prototype),
- **définir une valeur par défaut**, ou définir un ensemble de valeurs permises (frame prototype),
- **constituer un démon**, sorte de procédure qui peut être activée chaque fois que l'attribut doit être modifié,
- être un **pointeur sur un autre frame** (récursivité de frame), ce qui permet de représenter des objets composés

### • attributs et héritage :

- si la valeur d'un attribut est défini au niveau d'un prototype, tous les frames (prototypes et instances) plus spécifiques en héritent
- les instances héritent des méthodes définies au niveau prototype

## Attribut d'un Frame

Exemple de frame (tiré de SHIRKA de RECHENMANN) :



## Les aspects d'attributs de Frame

- un attribut est structurés par des aspects
- chaque aspect introduit une dimension décrivant l'attribut
- l'ensemble des aspects est fixé par le langage, c'est ce qui définit la sémantique de la représentation

- plusieurs types d'aspects peuvent être rencontrés :

- Aspect de typage
- Aspect d'obtention de valeur :
- Aspect réflexe
- Aspect de contrôle
- Aspect de communication homme-machine
- ...

## Les aspects d'attributs de Frame

- **Aspect de typage** : permet de définir le type de valeur des attributs d'un frame (valeurs permises) :

- type simple: entier, booléen, chaîne de caractères, ...
- type composé: un autre frame de telle nature, frame "personne" dans l'exemple précédent.

peut aussi donner une indication de restriction de typage :

- sur un intervalle de valeur; aspect "**intervalle**",
- une nono-valuation ou multi-valuation; aspect "**un**" et "**liste\_de**",
- l'aspect "**à\_vérifier**" introduisant une procédure à satisfaire pour toute valeur,

- **Aspect d'obtention de valeur** : permet de spécifier des procédures d'obtention d'une valeur :

- aspect "**valeur**": introduit la valeur de l'attribut
- aspect "**si\_besoin**": introduit une méthode de calcul de la valeur
- aspect "**défaut**": introduit une valeur par défaut (traitement des exceptions).

## Les aspects d'attributs de Frame

- **Aspect reflexe** :

permet d'introduire des procédures déclenchées en cas d'ajout, de modification, de suppression d'informations, ceci notamment pour des soucis de maintien de cohérence :

- aspect "**si\_modif**", "**si\_ajout**", "**si\_supprime**",

- **Aspect de contrôle** :

permet d'introduire des procédures définissant des actions à déclencher en cas d'échec ou de succès d'obtention de valeur, pouvant conduire à une propagation dans d'autres frames. Par exemple:

- aspect "**si\_succès**", "**si\_échec**", ...

- **Aspect de communication Homme-machine** :

permet d'introduire des procédures permettant de passer d'une vision interne à une vision externe des valeurs ou réciproquement :

- aspect "**lire**", "**écrire**", ...

## Héritage dans les frames : héritage simple

• les frames s'inscrivent dans une structure de demi-treillis → un frame peut dominer un ensemble de frames plus spécifiques et un frame peut être dominé par un ou plusieurs frames plus généraux.

• 2 situations d'héritage simple :



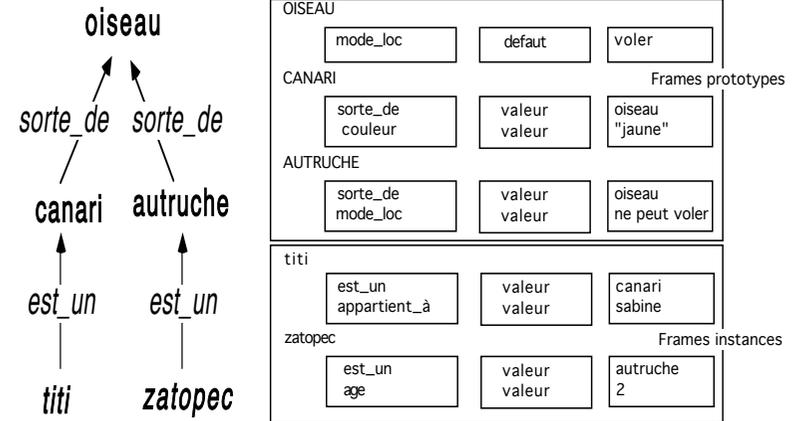
- d'un frame instance de son frame prototype (relié par un attribut "est\_un")
- d'un frame prototype d'un autre frame prototype (relié par un attribut "sorte\_de")

• d'une façon générale :

- si la valeur de l'attribut est défini au niveau d'un **prototype**, tous les frames **prototypes et instances plus spécifiques en héritent**
- il y a aussi héritage des **procédures** ou **méthodes** définies au niveau des prototypes au niveau des instances.

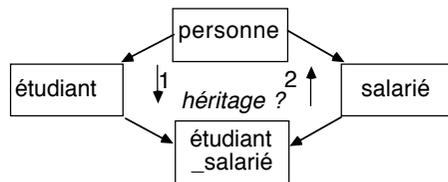
## Héritage dans les frames

### Héritage simple



## Héritage dans les frames : héritage multiples

- lorsqu'un schéma est dominé par plusieurs autres frames
- définir l'ordre d'application des héritages, par exemple par une liste de précédence ou une énumération des frames à héritage :



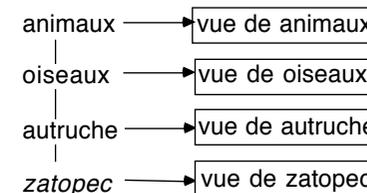
• sens de parcours possible pour l'héritage :

- 1) étudiant\_salaire  
étudiant  
personne  
salaire
- 2) étudiant\_salaire  
salaire  
personne

## Inférence d'instanciation dans les frames

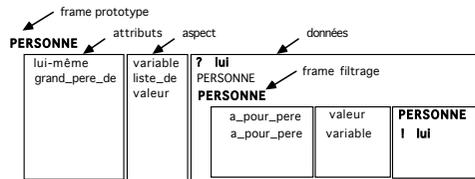
- l'instanciation de frames de type instance constitue l'inférence **fondamentale** dans les systèmes de frames
- consiste à "**construire**" ou "**compléter**" une instance de frame = **instancier par des valeurs les attributs définis dans les frames prototypes** auxquels est associé l'instance de frame considérée
- cette obtention de valeurs peut être **directe**; valeurs d'attributs propres à l'instance, ou par héritage.
- Si l'on appelle "**vue**" l'ensemble des attributs que possède un frame
- une **instance de ce frame sera l'union de toutes les vues propres et héritées** :

### FRAMES



## Inférence par filtrage dans les frames

- intervient quand l'aspect valeur d'un frame prototype renvoie sur un autre frame
- ce dernier frame intervient alors comme un filtre :
  - décrivant les instances possibles : valeurs que cet attribut pourra prendre
  - contenant généralement des conditions supplémentaires à vérifier sur les valeurs de ce frame restreignant encore les instantiations possibles.



### conventions:

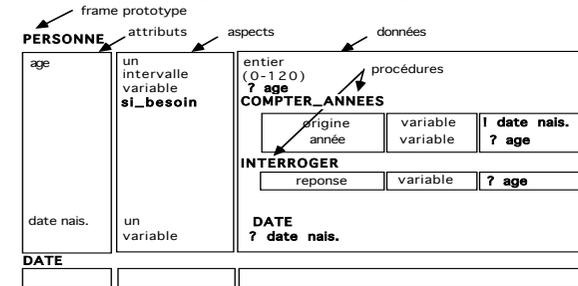
- dans "! X", le préfixe "! ..." indique que la valeur de la variable X doit être instanciée à l'appel du filtre, variable d'entrée,
- dans "? X", le préfixe "? ..." indique que la valeur de la variable X peut être inconnu à l'appel du filtre, variable de sortie,

### Dans cet exemple:

- l'attribut "lui\_même" du frame prototype principal représente le frame,
- l'aspect "variable" de cet attribut nomme une variable "? LUI" qui représente l'attribut en question,
- à l'instanciation, dès que l'attribut reçoit une valeur, cette valeur est affecté à la variable,
- le déclenchement du filtre peut conduire à un retour arrière.

## Inférence procédurale dans les frames

- obtenue par activation des procédures, méthodes associées aux attributs afin d'en obtenir des valeurs d'instanciation
- ces procédures sont elles-mêmes décrites par des frames prototypes, dont les attributs sont leurs valeurs d'entrée et de sortie
- c'est l'aspect "si-besoin" qui distingue un frame de filtrage d'une telle procédure :



Ex : l'âge est calculé à partir de la date de naissance "date nais." par la procédure COMPTER\_ANNEES. Si cette date de naissance n'est pas connue, on interroge l'utilisateur pour obtenir l'âge avec la procédure INTERROGER.

## Quelques implémentation de frames

De nombreuses implémentations de frames ont été réalisées en langage LISP ou Prolog

- **FRL** : "Frame Représentation Language" a été proposé par B.ROBERTS et I.GOLDSTEIN en 1977 au MIT (un précurseur, rudimentaire)
  - les frames entre eux sont organisés **hiérarchiquement**
  - une base de connaissances en FRL = {frames} dont les attributs peuvent avoir les aspects suivants: valeurs limites, valeurs par défaut, contraintes et procédures (développées en LISP) pouvant être déclenchées quand une valeur sera dépassée ou non dépassée, voire utilisée par un autre attribut.
- **KRL** : (D.BOBROW et T.WINOGRAD au Xerox PARC)
  - s'inspire directement des idées de M.MINSKY, est plus ambitieux que le précédent.
  - il dispose d'opération d'unification de frames qui peuvent être contrôlées par le concepteur de la base de connaissances
  - il possède aussi la notion de "contexte de croyance" pouvant servir à définir un "mécanisme d'attention" (attention focusing mechanism)
  - il inclut de la "self Knowledge" dans le base de connaissance en fournissant des descriptions de descriptions.

## Quelques implémentation de frames

- les "unit" de KRL:
  - toute entité conceptuelle à représenter est décrite par un frame appelé "unit" appartenant à une "catégories" (Il y a 7 catégories dans KRL) définissant son niveau d'abstraction: objet générique, spécialisé ou élémentaire
  - une unit est définie par un ensemble d'attributs, décrits par de descripteurs.
  - les concepts généraux sont décrits par des units appartenant aux catégories "basic" et "abstract" :
    - les catégories "basic" définissent des particules de connaissances en arbre d'héritage simple dont elles sont les racines, ces particules peuvent être instanciées

EX : définition de l'unit basic "CHAT":

```
Chat UNIT Basic
<SELF>
<nom (a string)>
<race (a Race) Persan; DEFAULT >
<age (an integer)>
<propriétaire (a string)>
```

- les catégories "abstract" définissent des particules de connaissances plus générales qui ne peuvent être spécialisées.

## Introduction aux Scripts

**Extension du concept de frame** (R.C.SCHANK et R.ABELSON), **scripts** = séquence d'événements dans un contexte particulier, associés au concept de **sénario**.

Ex. de script : le restaurant :

rôle: **client, serveur, chef cuisinier, caissier**  
scène1: entrée  
PTRANS entrer dans le restaurant  
ATTEND chercher des yeux les tables libres  
MBUILD choisir une table où s'asseoir  
PTRANS s'y rendre  
MOVE s'y assoir  
scène2: Passer commande  
ATRANS recevoir le menu  
MTRANS lire le menu  
MBUILD décider d'un plat  
MTRANS le commander au serveur  
scène3: Manger  
ATRANS recevoir la nourriture  
INGEST l'ingurgiter  
scène4: Sortie  
MTRANS demander la note  
ATRANS recevoir la note  
ATRANS donner un pourboire au serveur  
PTRANS se rendre au caissier  
PTRANS payer le caissier  
PTRANS sortir du restaurant

*Les verbes MTRANS, PTRANS sont des primitives, de la même nature que les dépendances conceptuelle définies par R.C.SCHANK dans les réseaux sémantiques.*

## Interprétation dans les scripts

**Exemple : le programme SAM (Script Applier Mechanism), développé par R.C.SCHANK et son équipe:**

- reçoit en entrée un texte, mis sous la forme de **dépendances conceptuelles**,
- **identifie le script** qui doit être utilisé, le **remplit** avec les données contenues dans le texte,
- Il peut aussi faire les **inférences** qui sont nécessaires par exemple pour compléter une chaîne causale entre différentes actions décrites dans le texte,
- Comme pour les frames, les scripts peuvent inférer de l'information manquante. SAM peut alors répondre à des questions sur le texte.