

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Jijel
Faculté des Sciences exactes et de l'informatique
Département d'informatique



– Module –
Systemes Experts

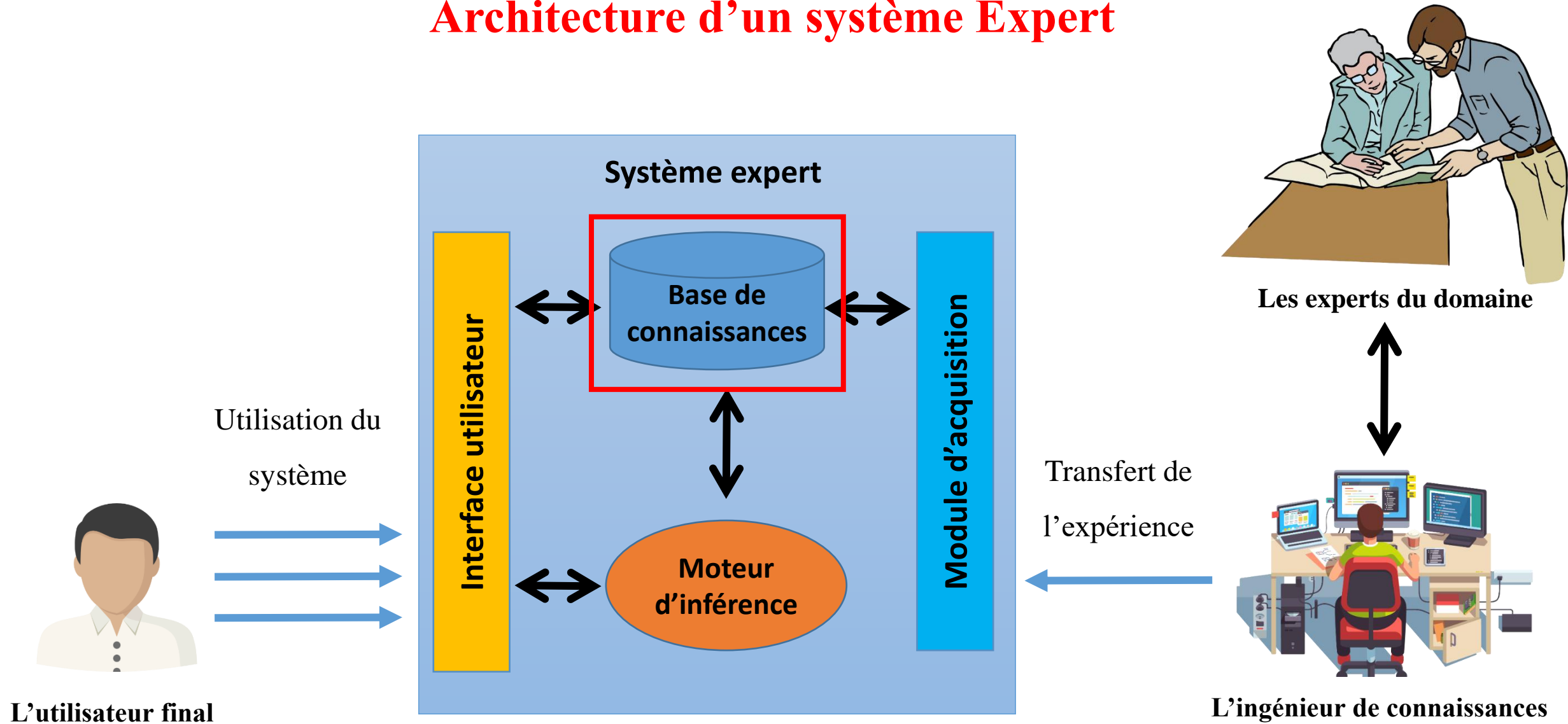
Master 1 : SIAD

Enseignant du module : Dr. Hemza FICEL

Contact: hemza.ficel@univ-jijel.dz

Chapitre 2 – Représentation de connaissances

Architecture d'un système Expert



Représentation logique

- ✚ Premier formalisme de représentation des connaissances en intelligence artificielle.
- ✚ Un formalisme qui permet de représenter des connaissances à l'aide de symboles.
- ✚ Une **formule** (expression/phrased) est un objet **syntactique** associé à une **signification sémantique** (interprétation).
- ✚ Deux concepts fondamentaux :
 - **la syntaxe** : suite de mots et de symboles formant une phrase.
 - **la sémantique** : la signification d'une phrase/sa valeur de vérité.
- ✚ Types de logique : logique propositionnelle, logique des prédicats, floue, ...

✚ **Logique propositionnelle** : une suite de symboles séparés par des conjonctions (et), des disjonctions (ou) ou des négations (non) ;

✚ **Logique des prédicats** : une suite de **symboles**, de **variables** et de **relations** avec des quantificateurs universels et existentiels (\forall, \exists) ;

✚ **Logique floue**: une extension de la logique classique qui **accompagne les faits de valeurs de vérité** (au lieu d'être **vrai** ou **faux**, les valeurs de vérité sont des réels **entre 0 et 1**).

Représentation logique: logique des prédicats (logique du 1er ordre)

Limites de la logique des propositions

La logique des propositions permet de représenter la situation suivante :

Si Ali est malade, il ne sort pas

P : Ali est malade

Q : Ali sort

$P \rightarrow \neg Q$

Ali est malade

P : Ali est malade

P

Ali ne sors pas

$\neg Q$: Ali ne sort pas

$\neg Q$

Pourquoi ? car elle voit la situation précédente comme **une combinaison** de plusieurs propositions simples (**P et Q**).

Limites de la logique des propositions

Est-ce que la logique des propositions permet de représenter la situation suivante ?

Si un homme est malade, il ne sort pas

P : Un homme est malade

Q : L'homme sort

$P \rightarrow \neg Q$

Ali est un homme malade

R : Ali est un homme malade

R

Ali ne sors pas

$\neg Q'$: Ali ne sort pas

$\neg Q'$

Non, car **les quatre propositions de cet exemple sont toutes différentes**, et le schéma de raisonnement précédent ne peut plus s'appliquer.

✚ La logique des prédicats est une extension de la logique des propositions qui permet de former une **représentation plus précise et expressive des connaissances** dans un domaine spécifique.

✚ **Pour quelle raison ?** Contrairement à la logique des propositions, qui ne peut représenter que des propositions simples (vrai/faux), la logique des prédicats permet de **représenter des relations plus complexes entre les objets** d'un domaine donné. Ainsi, elle permet de représenter des inférences valides qu'on ne peut pas représenter en logique des propositions (**P. ex., le raisonnement dans l'exemple précédent**).

✚ Les connecteurs logiques sont les mêmes qu'en logique des propositions

✚ **Nouveaux concepts** : termes (constante, variable et fonction), quantificateur existentiel et quantificateur universel.

Concepts de base : **Termes**

Concepts de base : **Termes**

✚ **Les termes** sont les éléments de base de la logique des prédicats. il existe trois types de termes : les constantes, les variables et les fonctions.



Exemples de termes en logique des prédicats :

« **Ali** », « **Mohammed** »; **x,y**; **homme(x)**, **père(x,y)**

Concepts de base : **Termes**

✚ **Les constantes** (notées en majuscules) sont des termes qui représentent **des objets spécifiques** du domaine ciblé. Elles peuvent être des personnes, des objets physiques, des lieux, des valeurs numériques, etc.



« **Ali** », « **Mohammed** » sont des exemples de **constantes** en logique des prédicats.

« **Jijel** », « **Algérie** » sont des exemples de **constantes** en logique des prédicats.

« **18** », « **23** », « **50** » sont des exemples de **constantes** en logique des prédicats.

Concepts de base : Termes

✚ **Les constantes** (notées en majuscules) sont des termes qui représentent **des objets spécifiques** du domaine ciblé. Elles peuvent être des personnes, des objets physiques, des lieux, des valeurs numériques, etc.

Les constantes sont souvent représentées par **des lettres majuscules**



PI = 3,1415926535897..... représente **la constante** mathématique pi.

A = ALI représente **un objet (une personne) spécifique**

Concepts de base : Termes

✚ **Les variables** (notées en minuscules) sont des termes qui représentent **des objets génériques** du domaine ciblé. Elles peuvent prendre des valeurs différentes selon la situation courante.

Les variables sont souvent représentées **par des lettres minuscules**



n : représente un entier positif **qui peut prendre comme valeur** : 0,1,2,4, ...

o : représente **un objet quelconque** d'un domaine donné. (p. ex, **o = Ali** ou **o = Alger**).

Concepts de base : Termes

✚ **Les fonctions (prédicats)** sont des termes qui représentent **des relations entre les objets** du domaine ciblé. Elles sont généralement utilisées pour construire des **termes plus complexes** à partir de constantes et de variables.

✚ **P : Ali est un homme.**

✚ **Q: Mohammed est un étudiant.**

✚ **R : Ali marche.**

✚ **S : Le perroquet vole dans l'air**

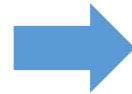
Dans chaque proposition on peut identifier :

- Une partie « **propriété/prédicat** » (**symbole de fonction**) : Homme, étudiant, marche, vole, ...
- Une partie « **argument/sujet** » (**symbole de constante individuelle ou variable**) : Ali, Mohammed, perroquet, ...

Concepts de base : Termes

✚ **Les fonctions (prédicats)** sont des termes qui représentent **des relations entre les objets** du domaine ciblé. Elles sont généralement utilisées pour construire des **termes plus complexes** à partir de constantes et de variables.

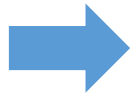
✚ **P** : Ali est un **homme**.



H = être un **homme**, **A** = Ali

H(A)

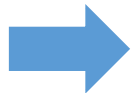
✚ **Q** : Mohammed est un **étudiant**.



E = être un **étudiant**, **M** = Mohammed

E(M)

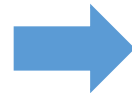
✚ **R** : Ali **marche**.



M = capable de **marcher**, **A** = Ali

M(A)

✚ **S** : Le perroquet **vole** dans l'air.



O = capable de **voler**, **p** = perroquet

O(p)

Concepts de base : Termes

✚ **Les fonctions (prédicats)** sont des termes qui représentent **des relations entre les objets** du domaine ciblé. Elles sont généralement utilisées pour construire des **termes plus complexes** à partir de constantes et de variables.

H = être un **homme**, **A** = Ali

H(A)

O = capable de **voler**, **p** = perroquet

O(p)



✚ On peut clairement voir qu'une **proposition** est considérée comme l'application d'un **prédicat (fonction)** à un **sujet (constante ou variable)**.

Concepts de base : Termes

✚ **Les fonctions (prédicats)** sont des termes qui représentent **des relations entre les objets** du domaine ciblé. Elles sont généralement utilisées pour construire des **termes plus complexes** à partir de constantes et de variables.

H = être un **homme**, **A** = **Ali**

H(A)

O = capable de **voler**, **p** = **perroquet**

O(p)

Un **sujet peut** être une constante lorsqu'on souhaite parler d'une entité particulière (p. ex., « Ali » pour parler d'une personne nommée Ali.)

Un **sujet peut** être une variable lorsqu'on veut parler de n'importe quelle entité d'un certain type, (p. ex., « p » pour parler de n'importe quel **perroquet**).

Concepts de base : Termes

✚ **Les fonctions (prédicats)** sont des termes qui représentent **des relations entre les objets** du domaine ciblé. Elles sont généralement utilisées pour construire des **termes plus complexes** à partir de constantes et de variables.

✚ **P : Ali est plus grand que Mohammed**



G = est plus grand, A = Ali, M = Mohammed

Prédicat binaire

G(A,M)

On peut donc facilement trouver le schéma d'inférence

Ali est plus grand que Mohammed **G(A,M)**

Mohammed est plus grand que Walid **G(M,W)**

Ali est plus grand que Walid **G(A,W)**

Concepts de base : Termes

✚ **Les fonctions (prédicats)** sont des termes qui représentent **des relations entre les objets** du domaine ciblé. Elles sont généralement utilisées pour construire des **termes plus complexes** à partir de constantes et de variables.

G = est plus grand, **A** = Ali, **M** = Mohammed

G(A,M) Ali est plus grand que Mohammed

G(M,A) Mohammed est plus grand que ALI



L'ordre des sujets (arguments) **est très important.**

Concepts de base : Quantificateur

Concepts de base : Quantificateur

Les notions précédentes permettent de représenter la proposition suivante

 **P : Ali est un étudiant malade**

EM = être malade, ET = être étudiant A = Ali

$EM(A) \wedge ET(A)$

Comment représenter la proposition suivante ?

 **Q : Tous les étudiants sont malades**

$EM(t) \wedge ET(t) \text{ ?????}$

Concepts de base : **Quantificateur**

Pour répondre à la question précédente, il faut préciser si notre proposition s'applique à **toutes** les entités, **à au moins** une entité, **ou à aucune entité** ?



✚ **Les quantificateurs** sont des symboles utilisés dans la logique des prédicats pour préciser la portée d'une proposition sur un ensemble d'entités.

Concepts de base : Quantificateur

La proposition concerne une valeur **singulière**
(c.-à-d. elle ne concerne que Ali)

⊕ P : Ali est un étudiant malade

EM = être malade, ET = être étudiant A = Ali

$EM(A) \wedge ET(A)$

La proposition Q concerne une valeur **universelle**
(c.-à-d. elle concerne tous les hommes)
Il n'y a pas de sujet individuel comme dans la proposition P

⊕ Q : Tous les étudiants sont malades

Concepts de base : Quantificateur



Pour représenter la proposition précédente, on doit introduire : (1) **une variable** et (2) **un signe** pour dire « **pour toute valeur de cette variable** ».

✚ Q : Tous les **étudiants** sont **malades**

EM = être **malade**, ET = être étudiant, A = Ali

Pour **tout x**, si x est un **étudiant**, alors x est **malade**"

$\forall x, (EM(x) \rightarrow ET(x))$

(2) Quantificateur **universel**
« pour toute valeur de la variable x ».

(1) la variable

Concepts de base : Quantificateur

✚ **Le quantificateur universel (\forall)** : est un symbole qui permet d'exprimer qu'une proposition s'applique à toutes les entités d'un ensemble donné.

✚ **Tous les chats sont des animaux**

$\forall x, (\text{chat}(x) \rightarrow \text{animal}(x))$

✚ **Le quantificateur existentiel (\exists)** : est un symbole qui permet d'exprimer **qu'il existe au moins une entité** dans un ensemble donné pour laquelle une proposition est vraie.

✚ **Il existe au moins un chat qui est noir.**

$\exists x, (\text{chat}(x) \wedge \text{noir}(x))$

Il existe un x tel que x est un chat et x est noir

Concepts de base : **Connecteurs**

Concepts de base : **Connecteurs**

✚ Le connecteur unaire : \neg (**non**)

✚ Les connecteurs binaires :

- \wedge (**et**) ;
- \vee (**ou**) ;
- \rightarrow (**implique**) ;
- \leftrightarrow (**équivalent**).

✚ Les parenthèses : ().

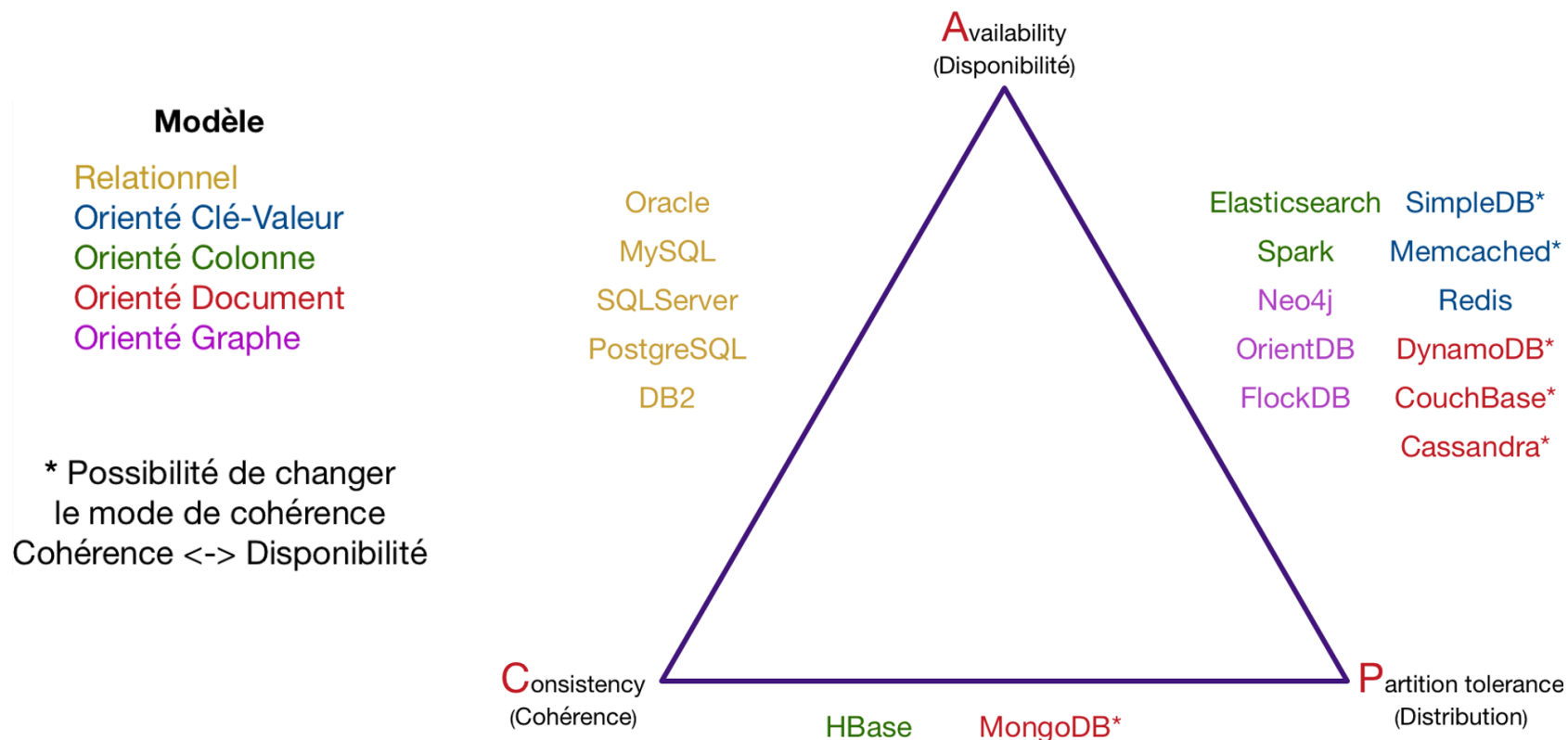
✚ Le quantificateur universel \forall , Le quantificateur existentiel \exists



LE SAVIEZ-VOUS



✚ Théorème de CAP (Brewer) : il est impossible dans une base de données de respecter en même temps les trois contraintes suivantes : la cohérence (**C**onsistency), la disponibilité (**A**vailability) et la distribution (**P**artition).



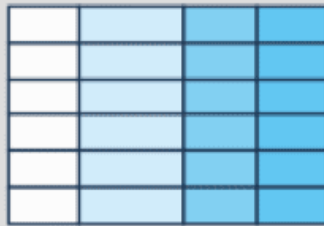


LE SAVIEZ-VOUS

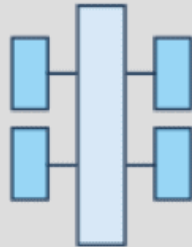


SQL

Relational

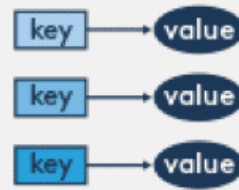


Analytical (OLAP)



NoSQL

Key-Value



Column-Family



Graph



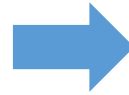
Document



Exercice

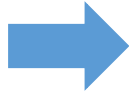
Traduire, aussi précisément que possible, les phrases suivantes en logique des prédicats.

✚ Tout est relatif



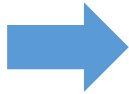
$\forall x, \text{relatif}(x)$

✚ Une porte est ouverte ou fermée.



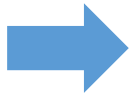
$\forall x, (\text{porte}(x) \rightarrow (\text{ouvert}(x) \vee \text{fermé}(x)))$

✚ Pour tout entier il existe un entier plus grand.



$\forall x, (\text{entier}(x) \rightarrow \exists y (\text{entier}(y) \wedge \text{plus-grand}(y,x)))$

✚ Certains oiseaux volent



$\exists x, (\text{oiseau}(x) \wedge \text{vole}(x))$

Exercice

Traduire, aussi précisément que possible, les phrases suivantes en logique des prédicats.

✚ Ali suit un cours.

✚ $\exists x, (\text{cours}(x) \wedge \text{suit}(\text{Ali}, x)).$

✚ Un cheval est plus rapide qu'un chien.

✚ $:\forall x \forall y (\text{cheval}(x) \wedge \text{chien}(y) \rightarrow \text{rapide}(x, y))$

✚ Un entier naturel est pair ou impair

✚ $\forall n(\text{naturel}(n) \rightarrow \text{pair}(n) \vee \text{impair}(n))$

✚ Un enseignant-chercheur a toujours un nouveau sujet à étudier.

✚ $\forall x(\text{ec}(x) \rightarrow (\exists y (\text{nouveau}(y) \wedge \text{etudie}(x, y))))$

Exercice (examen 2022)

Traduire, aussi précisément que possible, les phrases suivantes en logique des prédicats.

✚ Ali et Mohammed réussiront leur examen s'ils sont forts en logique

✚ ?

✚ Tous les touristes qui visitent Jijel l'aiment

✚ $J = \text{Jijel}$

✚ $\forall x ((\text{Touriste}(x) \wedge \text{Visite}(x, J)) \rightarrow \text{Aime}(x, J))$

✚ Quand quelqu'un fait confiance à quelqu'un qui l'a trompé, il a tort

✚ $\forall x (\text{Personne}(x) \wedge \exists y (\text{Personne}(y) \wedge (\text{FaireConfiance}(x, y) \wedge \text{Trompe}(y, x))) \rightarrow \text{Tort}(x))$

Représentation procédurale: règles de production

Règles de production

✚ Les règles de production sont des déclarations conditionnelles qui spécifient une action ou une conclusion à prendre si une certaine condition est remplie

Une règle de production est généralement formulée de la manière suivante :



SI [condition] **ALORS** [action]

Règles de production

SI [condition] **ALORS** [action]

La partie "si" décrit les conditions sous lesquelles la règle s'applique (une ou plusieurs propositions logiques qui peuvent être des conditions simples ou plus complexes)

La partie "alors" décrit les actions à entreprendre si la règle est déclenchée (si les conditions de la partie "si" sont remplies).

Règles de production

Exemple : la base de règles du système expert « pâtisserie » :

Base de règles
R1 : Si farine et beurre et œufs et sel alors pâte
R2 : Si pommes et sucré alors pommes sucrées
R3 : Si pommes sucrées et pâte alors tarte aux pommes
R4 : Si abricots et pâte alors tarte aux abricots
R5 : Si poires et pâte alors tarte aux poires
R6 : Si cerises et pâte alors tarte aux cerises

Règles de production

Exemple : la base de règles d'un système expert pour la recommandation de produits en ligne

Base de règles
R1 : SI l'utilisateur a acheté un produit A ET un produit B ET un produit C ALORS recommander le produit D .
R2 : SI l'utilisateur a acheté le produit E ET le produit F ET le produit G ALORS recommander le produit H .
R3 : SI l'utilisateur a acheté le produit A ET le produit E ALORS recommander le produit I .
R4 : SI l'utilisateur est anonymes ALORS recommander le produit L .
R5 : Si le budget est supérieur à 1000 euros ALORS recommander le produit K .

Règles de production

Exemple : la base de règles d'un système expert visant à organiser les produits dans une supérette

Base de règles
R1 : Si l'utilisateur achète du café et du sucré , ALORS il est probable qu'il achète aussi du lait .
R2 : Si le client achète du lait , ALORS il est probable qu'il achète également du pain .
R3 : Si le client achète de la levure chimique ET des œufs , ALORS il est probable qu'il achète également de la farine .
R4 : Si le client achète du pain ET du lait , alors il est probable qu'il achète également du beurre
R5 : Si le client achète du café ET du lait , alors il est probable qu'il achète aussi du sucré

Pas de formalisme universel ,
quel schéma de représentation

choisir ?



- ✚ La plupart des systèmes experts utilisent **les règles de production**, pourquoi ?
- ✚ **Les règles de production** sont formulées dans un langage proche du langage naturel qui est plus accessibles pour les utilisateurs qui ne sont pas des experts en logique ou en informatique.
- ✚ **Les règles de production** permettent aussi une plus grande transparence et une meilleure traçabilité du raisonnement utilisé.
- ✚ **Les règles de production** sont aussi plus flexibles et plus adaptables aux changements.

“

Cependant, le choix du mode de représentation des connaissances peut **avoir des implications très importantes** sur la fiabilité, la flexibilité, et les performances du système expert. Il est donc primordial de prendre en compte **plusieurs facteurs** avant de faire un choix ...

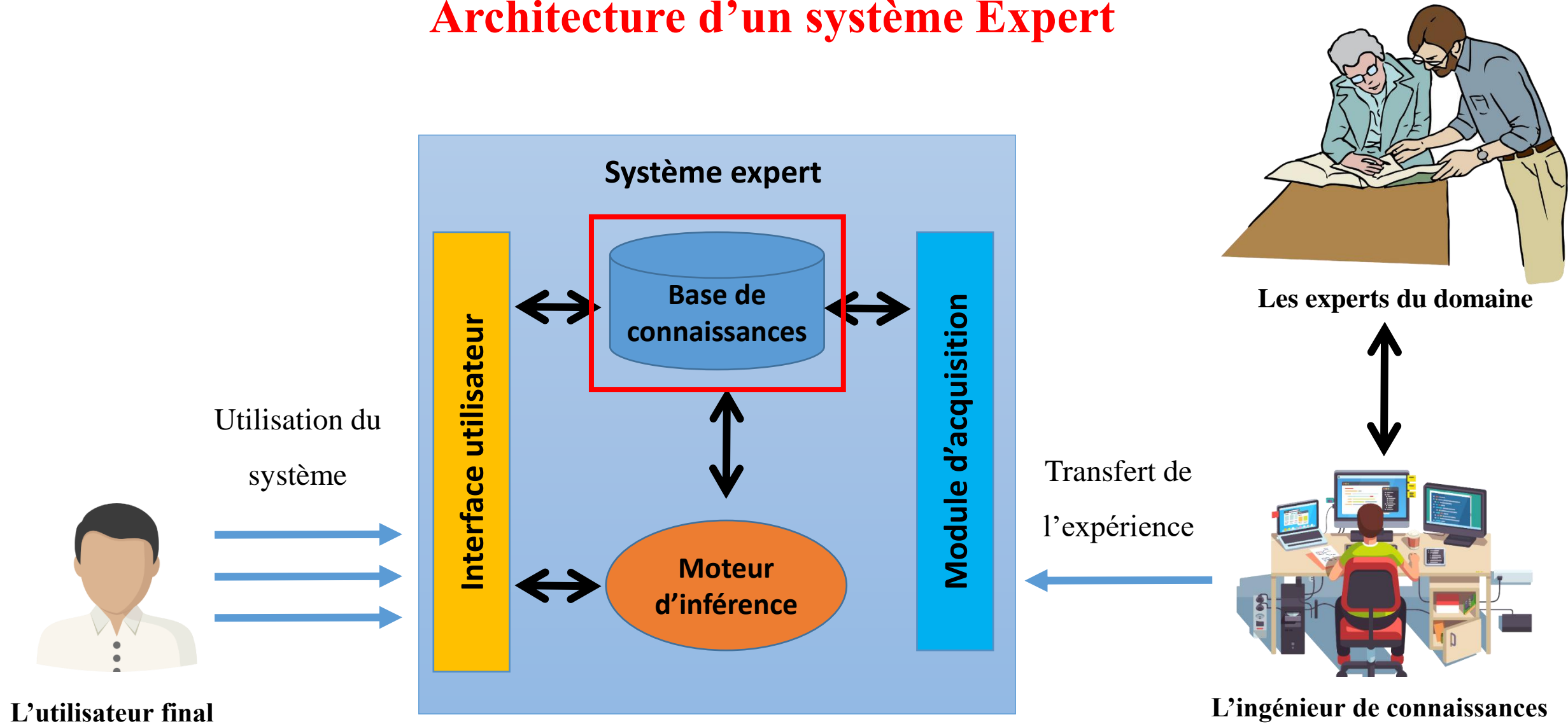
”

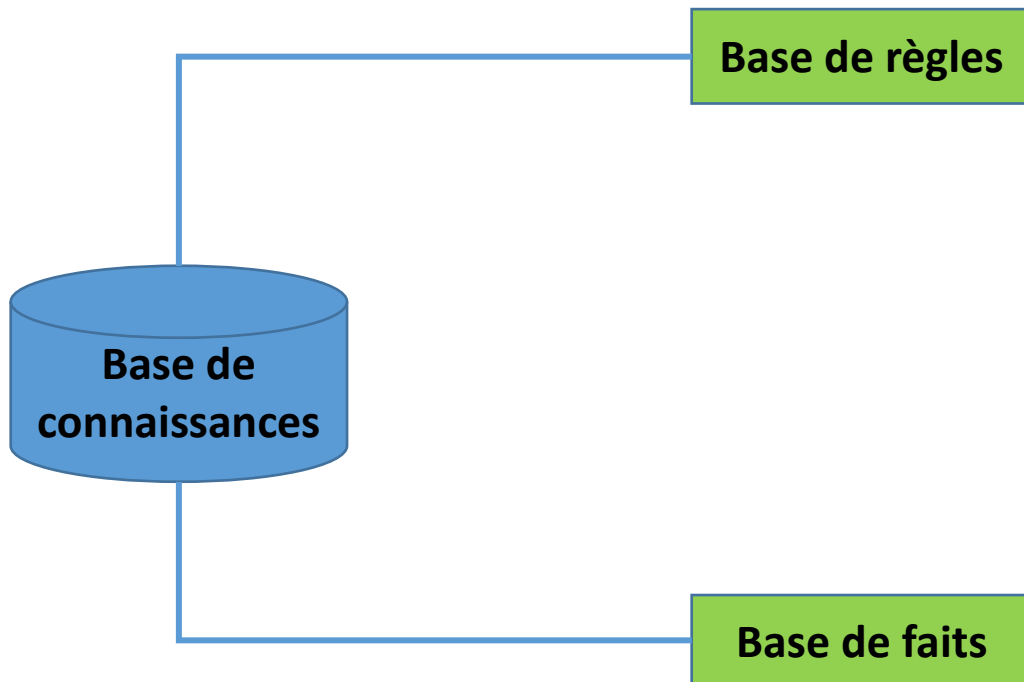
✚ **La nature et la complexité des connaissances à représenter** : pour représenter certaines connaissances, les règles de production sont le choix idéal, tandis que pour d'autres, des modes de représentation plus complexes, tels que la logique des prédicats, sont nécessaires pour représenter des connaissances impliquant des relations et des attributs complexes.

✚ **Les compétences des experts de domaine** : certains experts impliqués dans la conception du système peuvent être plus à l'aise avec certains modes de représentation qu'avec d'autres, ce qui peut influencer le choix du mode de représentation.

✚ **Les contraintes techniques** : les outils de développement du système et les contraintes techniques liées à la plateforme de déploiement peuvent aussi influencer le choix du mode de représentation.

Architecture d'un système Expert





- ✚ Conserve la connaissance sur l'**étude en cours** (constats, réalité).
- ✚ Son état évolue au cours de l'expertise.
- ✚ Les faits sont fournis par l'utilisateur qui dialogue avec le système.
- ✚ Considérée comme un espace de travail temporaire

- ✚ Conserve la connaissance sur le **domaine**.
- ✚ Les règles établissent des relations entre les faits.

Exemple : la base de connaissance du système expert « pâtisserie ».

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	Pommes , Poires Abricots, Farine Beurre, Sucre Sel
R2 : Si pommes et sucre alors pommes sucrées	
R3 : Si pommes sucrées et pâte alors tarte aux pommes	
R4 : Si abricots et pâte alors tarte aux abricots	
R5 : Si poires et pâte alors tarte aux poires	
R6 : Si cerises et pâte alors tarte aux cerises	