

INTERFACES HOMME-MACHINE

Sommaire

1) Problématique de la C.H.M

2) Modèles d'IHM

3) Méthode de conception d'une IHM

4) Ergonomie des IHM

5) Maquettage, prototypage, évaluation

6) Interfaces adaptatives, futur des IHM, conclusion

Un sondage au hasard

X

A quels domaines se rattache la CHM

X

Donnez les 3 modèles d'IHM ! Quelles relations entre eux ?

X

Quelle est la principale différence entre PAC et MVC?

Chapitre 3 : **Méthode de conception 00 d'une IHM**

3.1) Présentation de la méthode

3.2) Études des besoins pour la CHM

3.3) Spécifications de l'IHM

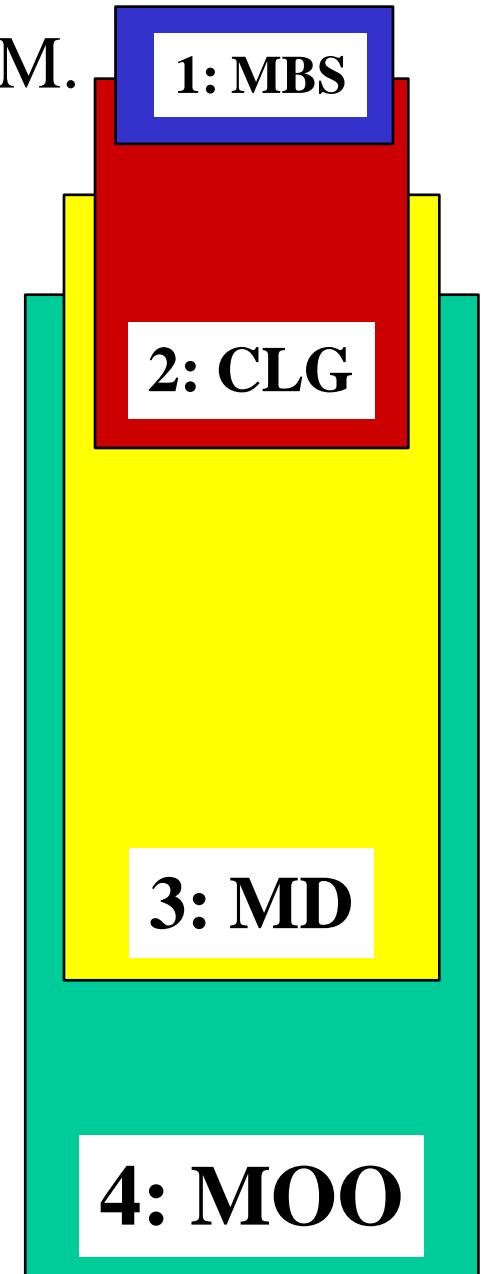
3.4) Réalisation et tests

3.1) Présentation de la méthode (1)

- ✓ Pas de méthode absolue de conception d'IHM.
 - plusieurs ont été proposées,
 - elles se reprennent les unes et les autres
- ✓ Quelques méthodes
 - 1^{ère} Méthode de Ben Schneidermann
 - Méthode «Common Language Grammar»
 - Méthode descendante de Ménadier
 - inspirée par MERISE

- Méthode orientée objet :
 - en continue évolution, pas encore figée

présentée dans ce cours



3.1) Présentation de la méthode (2)

- **Définir une interface utilisateur**
 - **n'est pas une opération cosmétique de dernière minute**
 - En amont, elle permet de valider les besoins
 - **c'est une opération fondamentale qui détermine**
 - la facilité d'utilisation et d'apprentissage
 - l'acceptabilité du système
 - la productivité.
 - **elle doit intégrer**
 - les règles ergonomiques (cf chap 4)
 - les caractéristiques des utilisateurs
- **La conception d'une IHM**
 - **est l'œuvre souvent d'équipes pluridisciplinaires**
 - **doit être réalisée en parallèle de la conception de l'application**
 - **doit être vérifiée au fur et à mesure**

3.1) Présentation de la méthode (3)

□ Règles méthodologiques générales

- **R1) Séparer la conception de l'interface de la conception du noyau de l'application**
 - Les méthodes de conception d'applications (*USDP, MERISE, ...*)
 - s'appuient sur une vue systémique de l'application
 - repoussent la conception de l'IHM au niveau de la réalisation.
 - La conception d'une interface part de **l'analyse de la tâche et de l'utilisateur.**
 - Ceci nécessite de définir une interface entre modèles applicatifs et modèles de la tâche
 - Dans certaines applications, pilotées par l'utilisateur, le dialogue est primordial, c'est lui qui structure l'application.

3.1) Présentation de la méthode (4)

Règles méthodologiques générales (suite)

□ **R2) Dans le cas des applications interactives, commencer à conception par l'interface**

C'est le dialogue qui structure l'application

➤ cela permet

- de construire très rapidement une maquette,
- de valider les choix ergonomiques et les spécifications,
- de faire prendre conscience de l'application à l'utilisateur et ainsi de le motiver.

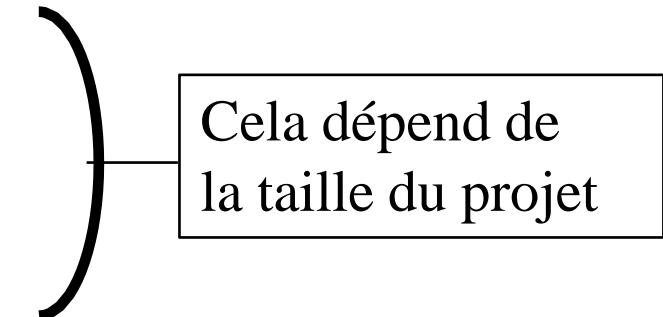
□ **R3) Prendre en compte tous les utilisateurs potentiels**

3.1) Présentation de la méthode (5)

Règles méthodologiques générales (suite)

R4) Créer une équipe pluridisciplinaire :

- concepteurs
- ergonomes
- spécialistes en communication
- graphistes
- utilisateurs (spécialistes de la tâche)



R5) Concevoir de manière interactive

- chaque phase doit être vérifiée expérimentalement

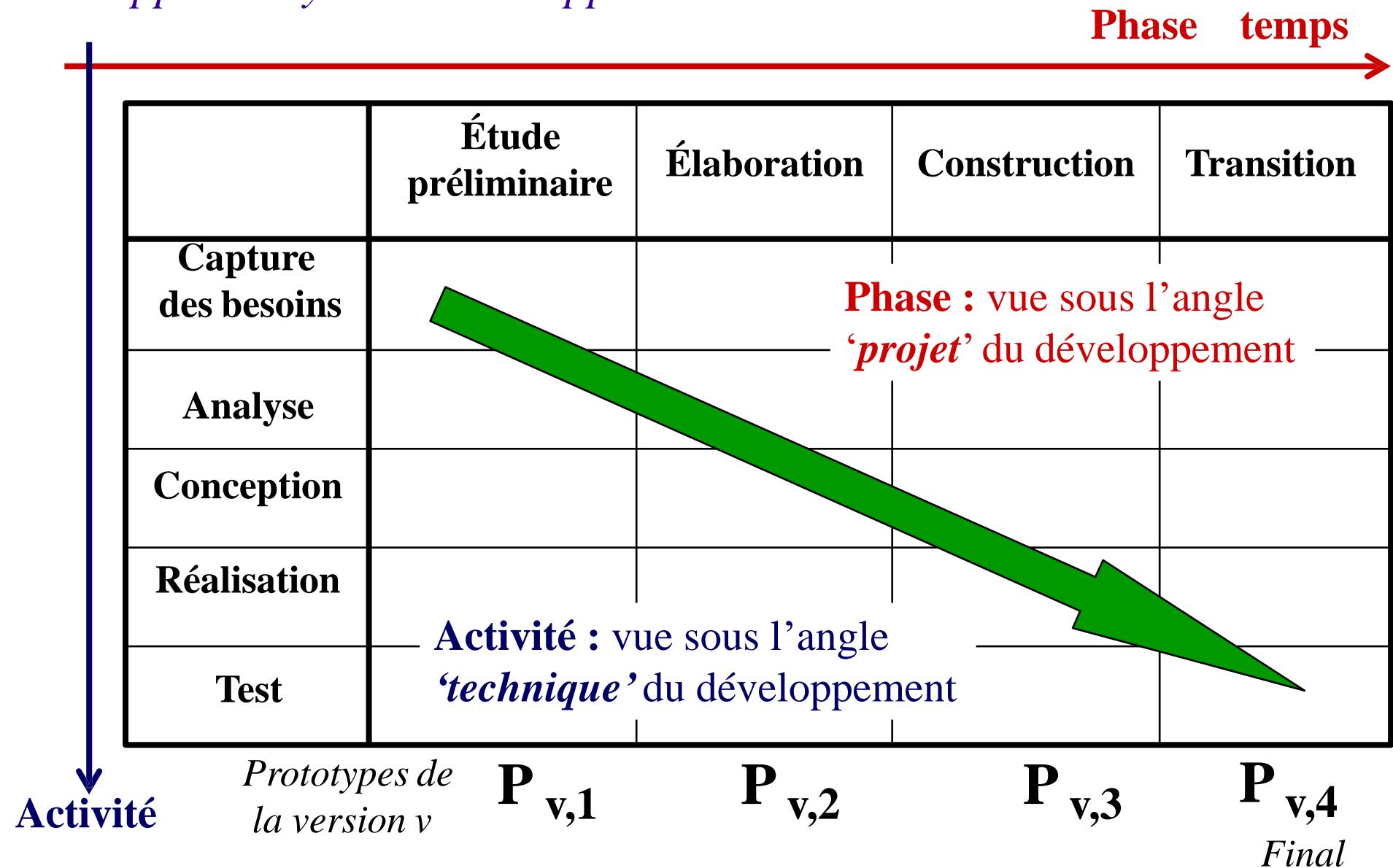
R6) Une méthode n'est pas une recette de cuisine

- chaque application est spécifique bon sens
- prévue pour les grandes application savoir élaguer

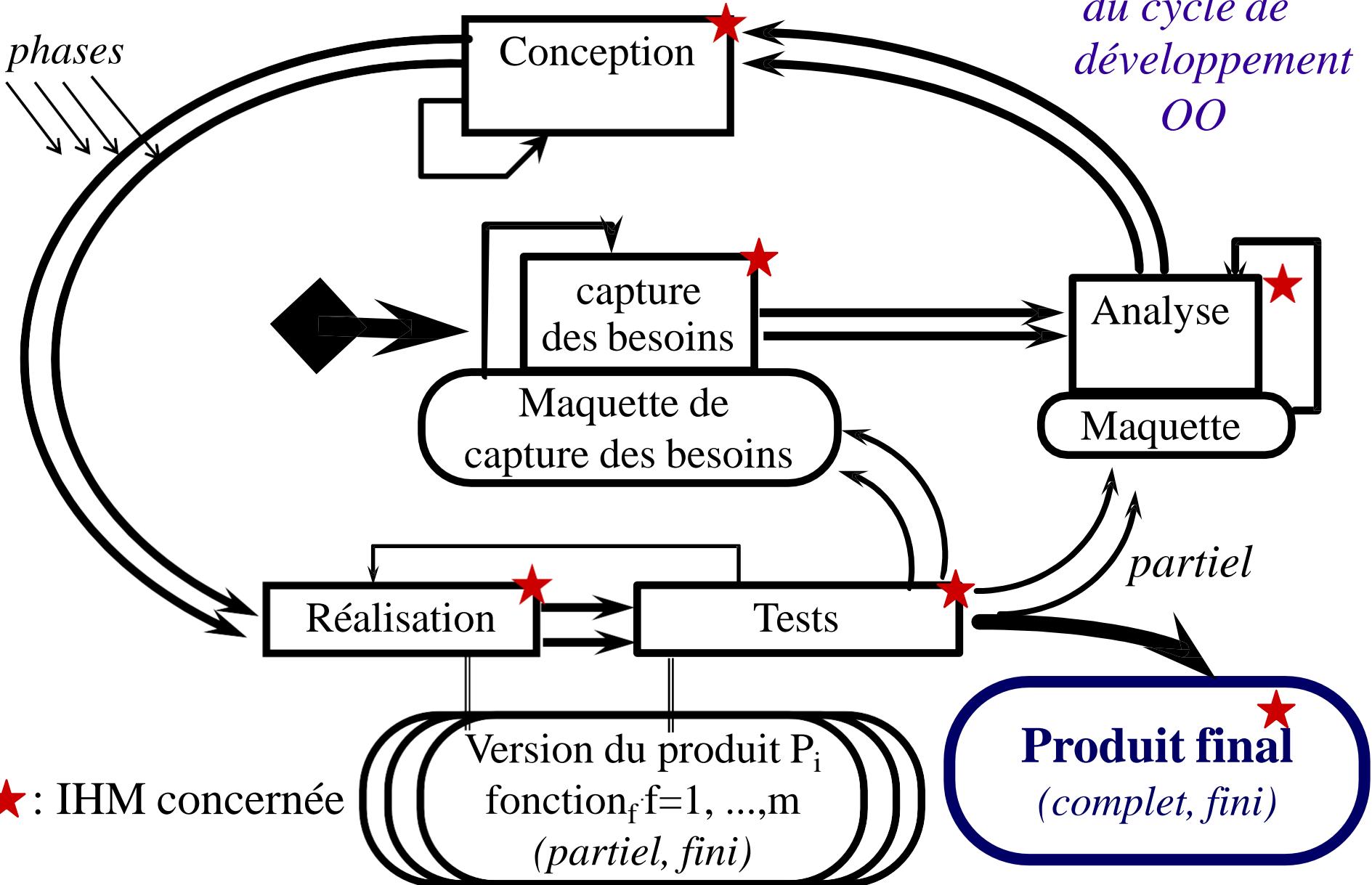
3.1) Présentation de la méthode (6)



Rappel du cycle de développement OO



3.1) Présentation de la méthode (7)



3.1) Présentation de la méthode (8)

Les phases

Phase 1 : étude préliminaire

Noyau de l'application

(rappel)

- Que fait le système ?
- A quoi pourrait ressembler l'architecture ?
- Quels sont les risques ?
- Quel est le coût estimé du projet ?
- Comment le planifier ?



IHM

- A qui s'adresse le système ?
- Quel type de communication ?
- Que pourront faire les utilisateurs ?
- Quelles sont les contraintes d'utilisation ?
- Quels sont les risques de rejet ?
- Coût et planification.

coût très faible

Accepter le projet ?



3.1) Présentation de la méthode (9)

Les phases

Phase 2 : élaboration

Noyau de l'application

(rappel)

- Spécification de la plupart des cas d'utilisation
- Conception de l'architecture à base.
- Mise en œuvre de cette architecture (cas critiques ~ 10 %)
- Planification complète

IHM

- Description des principales tâches des différents utilisateurs
- Principaux composants de l'IHM
 - type d'IHM et métaphore
 - principaux objets
 - principales commandes
 - le langage d'entrée - sortie
 - principales fenêtres avec les enchaînements
- Réalisation du squelette (~ 10 %)
- Planification complète

coût faible

Besoins, architecture, types et tâches des utilisateurs, choix ergonomiques, planning stable, risques contrôlés,

3.1) Présentation de la méthode (10)

Les phases

Phase 3 : Construction

Similaire pour NA et IHM

- Développement par incrément (NA et IHM)
 - L'architecture du NA et le squelette de l' IHM sont stables malgré des changements mineurs.
- Le produit contient tout ce qui a été planifié (NA et IHM)
 - Spécifications (NA et IHM) terminées
 - Conception (objets NA et IHM) terminée
 - Objets NA et IHM codés, testés
 - documentation rédigée
 - Intégration NA et IHM réalisée et testée

i l reste quelques erreurs

coût très fort
(> 50 %)

Produit suffisamment correct pour être installé chez un client (β -test)

3.1) Présentation de la méthode (11)

Les phases

Phase 4 : Transition

Similaire NA et IHM

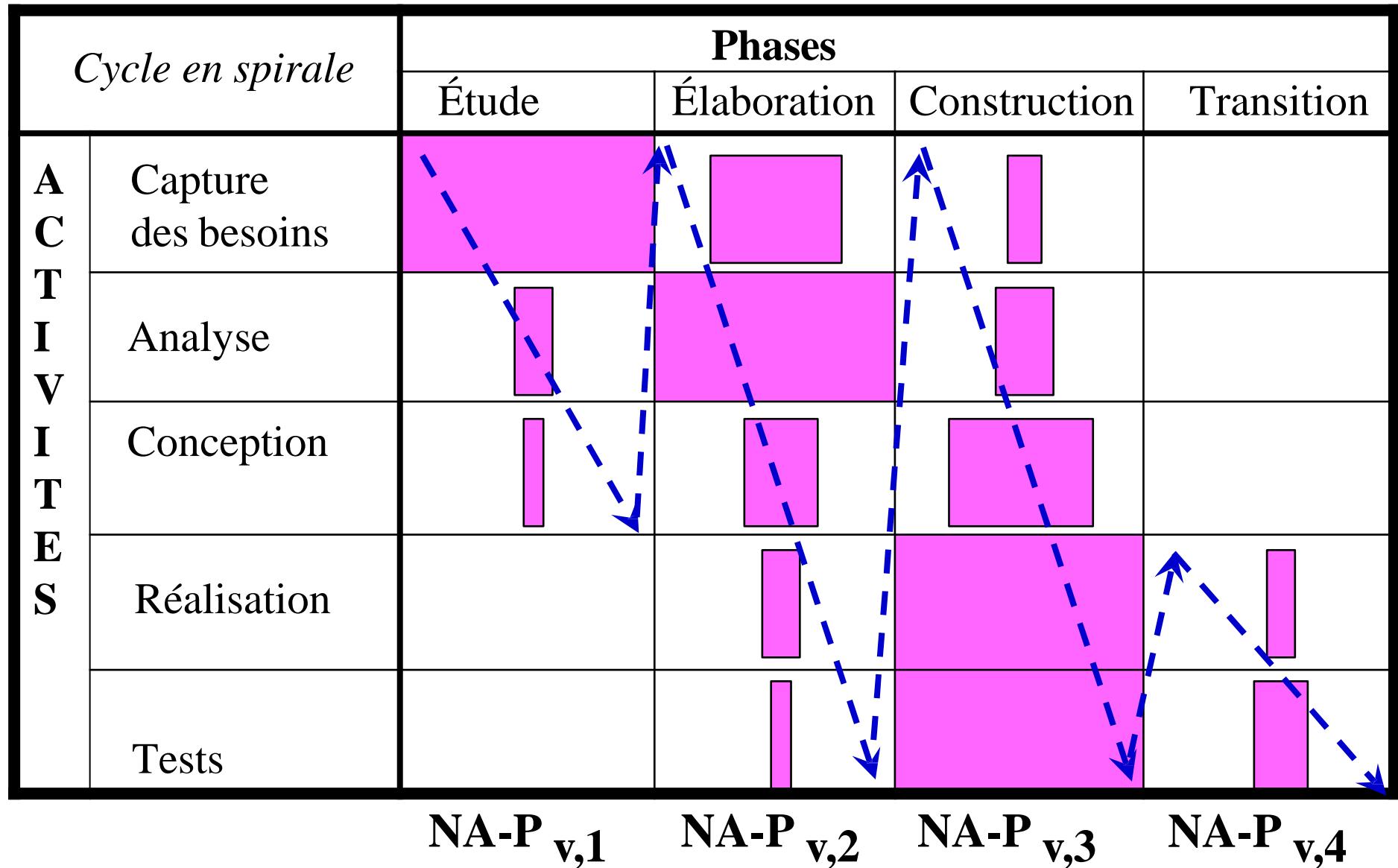
- Produit délivré (version β)
- Correction du reliquat d'erreurs
- Essai et amélioration du produit
- Formation des utilisateurs
- Installation de l'assistance en ligne
- Étude de la satisfaction des utilisateurs
- etc.



**Produit satisfaisant ?
modifications nécessaires ?**

3.1) Présentation de la méthode (12)

Cycle de développement USDP (*rappel*)

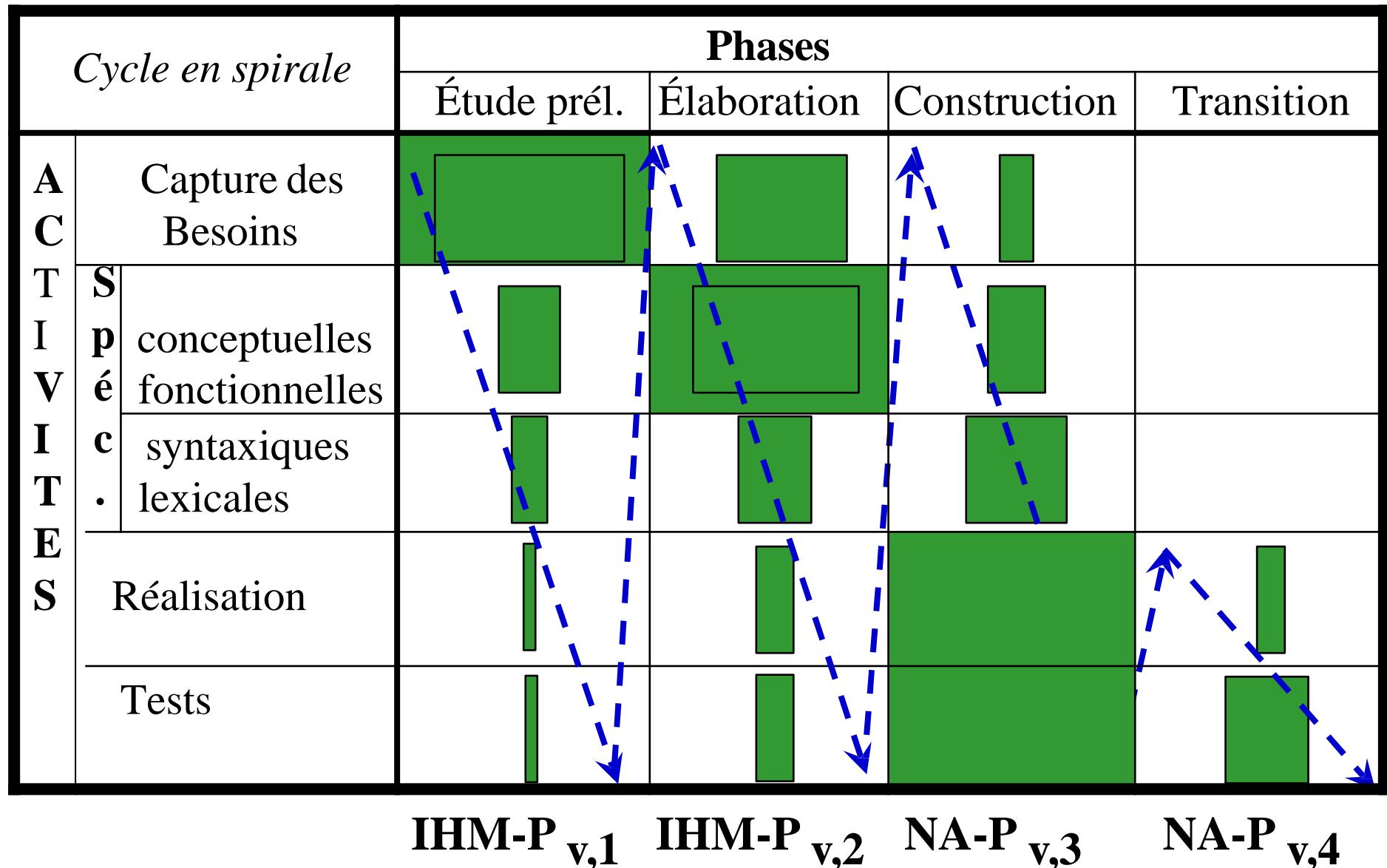


3.1) Présentation de la méthode (13)

Les activités	
Noyau de l'Application (UDSP)	IHM
1) besoins pour le NA	1) besoins pour la CHM
2) analyse	2) spécifications IHM <ul style="list-style-type: none">- conceptuelle- fonctionnelle
3) conception	<ul style="list-style-type: none">- syntaxique- lexicale
4) réalisation <i>incrémentale</i>	3) réalisation <i>incrémentale</i>
5) tests et validation	4) tests et validation
Intégration / validation	

3.1) Présentation de la méthode (14)

Cycle de développement IHM



3.1) Présentation de la méthode (15)

Exemple : La gestion du parc naturel européen du Mont Agalac

- *il s'agit d'un exemple fictif, simplifié et traité partiellement*
- ❖ Il s'agit de réaliser un logiciel de gestion d'un parc naturel européen.
 - **Ce parc est situé en région montagneuse.** Il occupe une superficie d'environ 10 000 KM². Il est découpé en région (~ 100), chaque région en zone (~ 100/région), chaque zone en parcelle (~ 1000/zone).
 - **La faune est importante et variée.** Elle comporte des animaux protégés et en nombre restreint. Ces animaux doivent être suivis (*santé, gestion*) régulièrement. Certains animaux rares ou/et pouvant être dangereux (*ex. ours*) doivent être localisés en permanence avec le plus de précision possible.

3.1) Présentation de la méthode (16)

□ Exemples (suite)

- **La flore est aussi importante et variée.** Elle comporte des espèces rares et protégées. Ces espèces doivent être suivies (*maladie, ...*).
- Certaines plantes pouvant être dangereuses (*ex. champignons*) et doivent être localisés en permanence avec le plus de précision possible.
- **Le parc accueille un public nombreux** et veut jouer un rôle éducatif et écologique. Plusieurs types de visites sont prévues : de la visite guidée en une journée à la randonnée libre de plusieurs jours. Les visiteurs s'engagent à respecter un contrat de visite :
 - respect de la flore et de la faune,
 - rester dans les sentiers balisés,
 - dormir dans les refuges (*réservation*) ou bivouaquer dans des terrains précis (*prévenir*),
 - participer à la vie et à l'amélioration du parc :
 - signaler toute anomalie (*plante ou animal malade, contrevenant*)
 - signaler toute nouveauté (*nouvel animal, nouvelle plante, ...*)

3.1) Présentation de la méthode (17)

Exemples (suite)

- **Objectifs du système :** couvrir tous les aspects de la vie du parc
 - sécurité (*des personnes, des animaux, du parc*),
 - gestion classique (*personnel, travaux*),
 - visiteurs (*accueil, informations, contrôle, etc.*),
 - faune, flore (*santé, statistiques, évolutions, etc.*)
 - recherches (*sur la faune, la flore, la géologie, etc.*)
 - etc.
- **Contraintes de réalisation :**
 - ✓ liaison WIFI couvrant la totalité du parc
 - ✓ Système pervasif en architecture client / serveur
 - ✓ Une grande variété de clients
 - ✓ Clients légers classiques dans les points d'accueil et les refuges gardés
 - ✓ Smartphone pour le personnel itinérant (gardes, guides, ...)
 - ✓ Téléphones portables standard pour les visiteurs : liaison sonore, textuelle (SMS), multimédia (MMS)
 - ✓ liaisons grand débit avec les autres parcs (*voisins, distants, etc.*) et les organismes de sécurité (*pompiers, gendarmerie, ...*)
 - ✓ serveurs INTERNET, INTRANET, etc.

Chapitre 3 : Méthode de conception 00 d'une IHM

3.1) Présentation de la méthode

➤ **3.2) Études des besoins pour la CHM**

✓ **3.2.1) Objectifs & sous activités**

3.2.2) Identification des domaines fonctionnels

3.2.3) Identification des classes d'utilisateurs

3.2.4) Analyse des tâches et de l'activité

3.2.5) Spécifications des performances d'usage

3.2.6) Synthèse

3.3) Spécifications de l'IHM

3.4) Réalisation et tests

3.2.1) Objectifs & sous activités

□ Définir les grands objectifs de l'application en terme de :

- **domaines fonctionnels et de profils d'utilisateurs**
 - *une identification des caractéristiques de la population d'utilisateurs potentiels.*
 - *une analyse des tâches & de l'activité de l'utilisateur.*
- **spécifications de performances d'usage**
 - *temps de réponse*
 - *facilité d'apprentissage*
 - *acceptabilité et utilisabilité, etc.*

□ 4 sous-activités

- *identification des domaines fonctionnels*
- *identification des classes d'utilisateurs (profils)*
- *analyse des tâches et de l'activité*
- *spécifications des performances d'usage*

Chapitre 3 : Méthode de conception 00 d'une IHM

3.1) Présentation de la méthode

➤ 3.2) Études des besoins pour la CHM

3.2.1) Objectifs & sous activités

✓ 3.2.2) Identification des domaines fonctionnels

3.2.3) Identification des classes d'utilisateurs

3.2.4) Analyse des tâches et de l'activité

3.2.5) Spécifications des performances d'usage

3.2.6) Synthèse

3.3) Spécifications de l'IHM

3.4) Réalisation et tests

3.2.2) Domaines fonctionnels (1)

- ✓ *différents*
- ✓ *rarement autonome*
- ✓ *nombre limité*

➤ Identification des domaines fonctionnels (DF)

- description macroscopique de l'activité de l'entreprise
- DF = regroupement de fonctions
- description hiérarchisée (notion de sous domaines)
- description des "feuilles" (textuelle ou/et liste de fonctions)
- indépendant des utilisateurs

livrable

DDF
*Description des
Domaines
Fonctionnels*

3.2.2) Domaines fonctionnels (2)

Exemple de DDF

Domaines Fonctionnels (DF)

D.1) Sécurité

D.2) Admin. du Parc

D.3) Admin. des visiteurs

D.i) -----

D.n) Admin. du système

Sous-Domaines Fonctionnels (SDF)

D.1.1) Sécurité du Parc

D.1.2) Sécurité des personnes

D.2.1) gestion du foncier

D.2.2) gestion de la flore

D.2.3) gestion de la faune

D.3.1) gestion des visiteurs

D.3.2) organisation des visites

D.n.1) admin-utilisateurs

D.n.2) maintenance

X

à vous de compléter

3.2.2) Domaines fonctionnels (3)

Exemple de DDF

Sous-Domaines Fonctionnels (SDF)

- D.1.1) Sécurité du Parc
- D.1.2) Sécurité des personnes
-
- D.2.1) gestion du foncier
- D.2.2) gestion de la flore
- D.2.3) gestion de la faune
-
- D.3.1) gestion des visiteurs
- D.3.2) organisation des visites
-
- D.n.1) admin-utilisateurs
- D.n.2) maintenance

tout ce concerne l'intégrité physique du Parc (protection contre les incendies, les inondations, etc.)

administration des parcelles, des chemins d'accès, des bâtiments, des clôtures, etc.

accueil, vente des tickets, localisation des visiteurs, fidélisation, etc.

X

à vous de compléter

Chapitre 3 : Méthode de conception 00 d'une IHM

3.1) Présentation de la méthode

➤ **3.2) Études des besoins pour la CHM**

- 1) *Objectifs & sous activités*
- 2) *Identification des domaines fonctionnels*

➤ **3.2.3) Identification des classes d'utilisateurs**

- 4) Analyse des tâches et de l'activité
- 5) Spécifications des performances d'usage
- 6) Synthèse

3.3) Spécifications de l'IHM 3.4)

Réalisation et tests

3.2.3) Classes d'utilisateurs (1)

- **Identification de profils d'utilisateurs**
 - ❖ Décrire les populations d'utilisateurs selon :
 - leur **expérience préalable** en ce qui concerne
 - la tâche à effectuer
 - l'outil informatique
 - afin de déterminer les dispositifs de communication
 la métaphore*
 - **l'homogénéité de la population**
 - débutant ou expérimenté dans la tâche / en informatique
 - afin de mettre au point des systèmes d'aide, de formation, ...*
 - **leur attitude générale vis à vis de l'informatique**
 - afin d'introduire le système et la formation*
 - **facteurs de personnalité**

3.2.3) Classes d'utilisateurs (2)

- *Identification de profils d'utilisateurs (suite)*

- ❖ les livrables :

- **Le modèle structurel des utilisateurs (MU)**

- les classes d'utilisateurs
 - les relations entre ces classe

utilisation du modèle de structure d'UML

- **Le graphe des profils utilisateurs (GPU)**

- un profil d'utilisateur une ou plusieurs classes utiles
 - un profil d'utilisateur une IHM
 - Profil (U2) hérite de Profil (U1)
 - IHM (U2) : composition (IHM (U1) + IHM-spé (U2)

utilisation du modèle de structure d'UML - attention c'est une vue purement logique et pas conceptuelle

3.2.3) Classes d'utilisateurs (3)

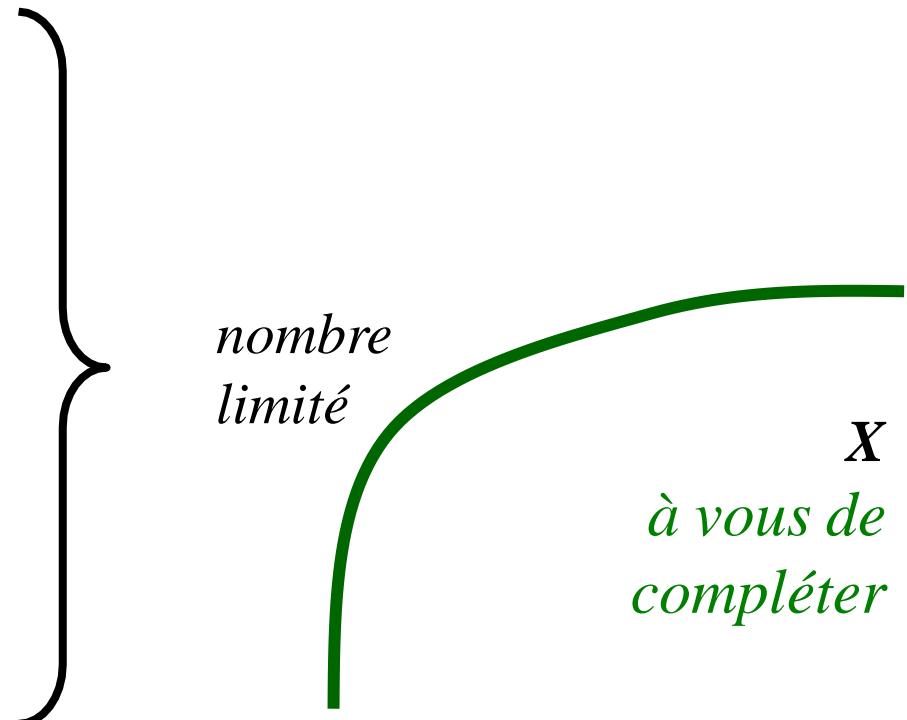
❖ les livrables (suite) :

➤ Description des profils utilisateurs (DPU)

- texte décrivant la population associée à ce profil
 - qui sont les utilisateurs et pas ce qu'il font,
 - principales caractéristiques (sexe, âge, niveau d'études, connaissance du métier, connaissance de l'informatique,)
 - nombre d'individus de cette classe et homogénéité du groupe
 - etc.

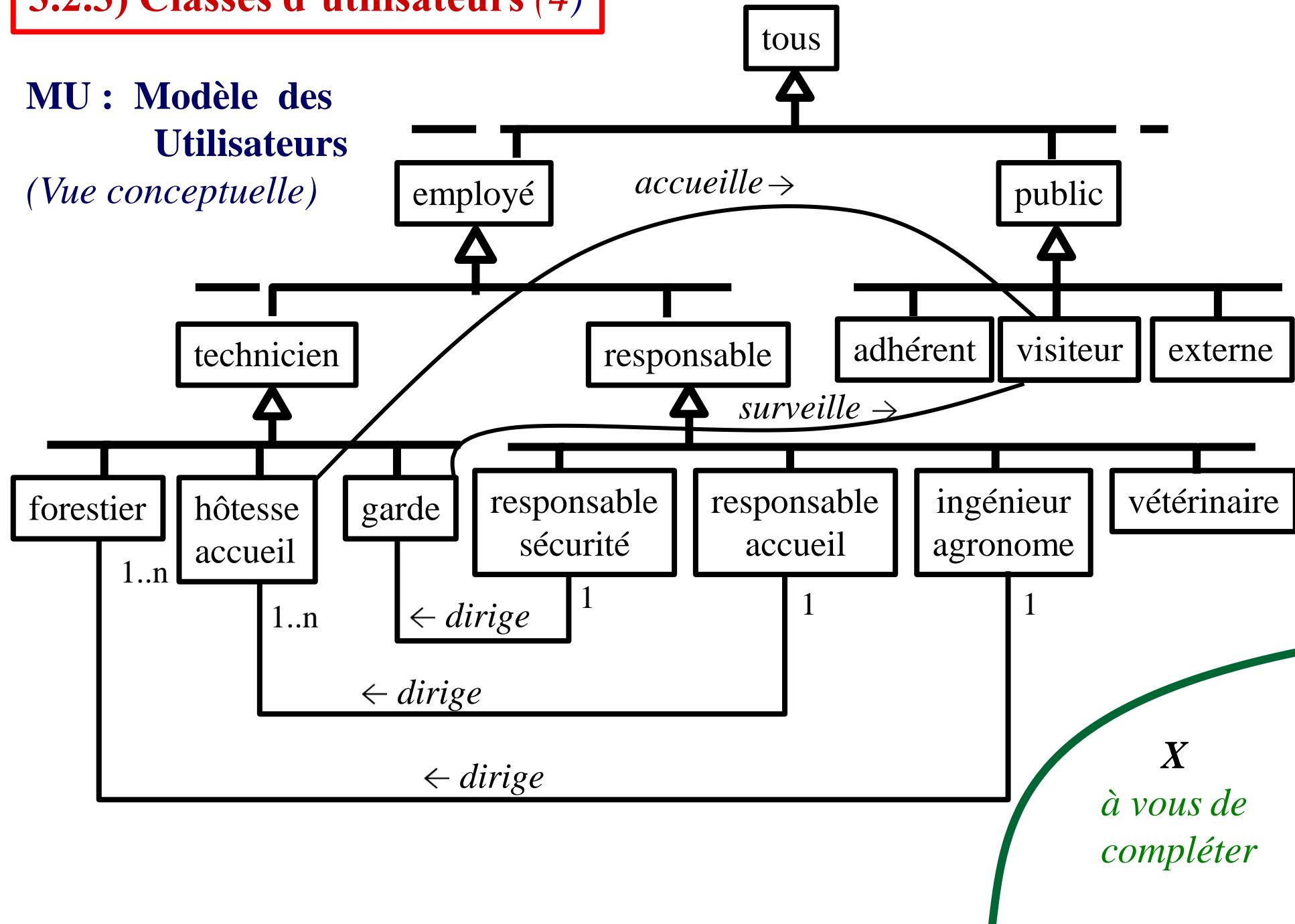
✓ Exemple de classes d'utilisateurs :

- *Garde*
- *Responsable -sécurité*
- *Hôtesse d'accueil*
- *Ingénieur agronome*
- *Vétérinaire*
- *Visiteur*
- *etc.*



3.2.3) Classes d'utilisateurs (4)

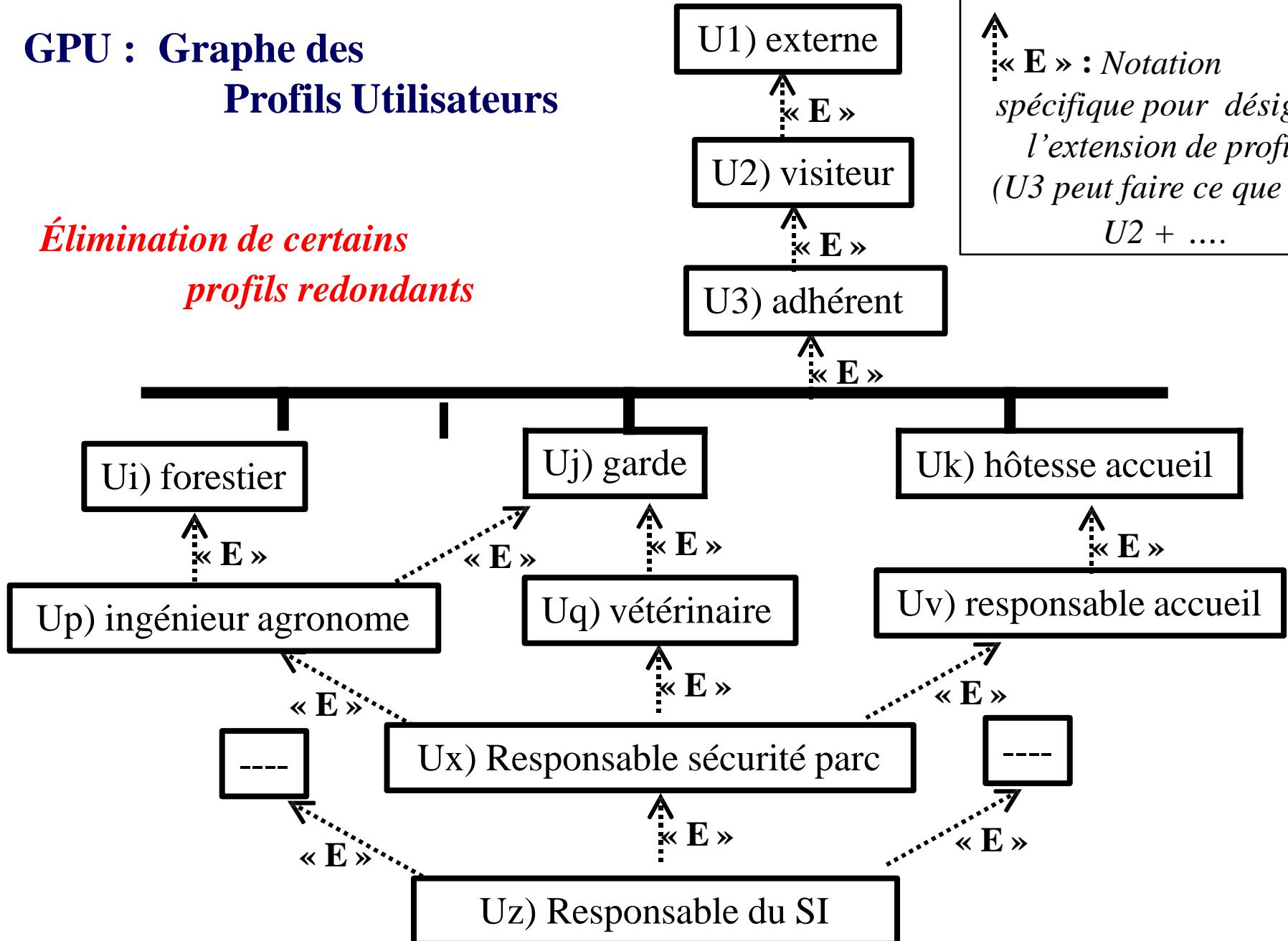
MU : Modèle des Utilisateurs
(Vue conceptuelle)



3.2.3) Classes d'utilisateurs (5)

GPU : Graphe des Profils Utilisateurs

Élimination de certains profils redondants

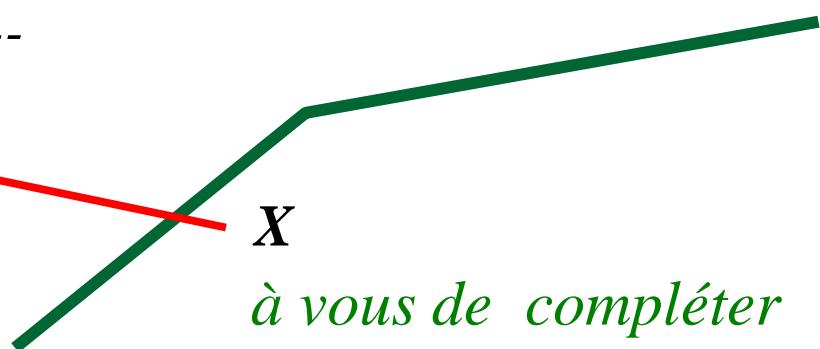


« E » : Notation spécifique pour désigner l'extension de profil
(U_3 peut faire ce que fait $U_2 + \dots$)

3.2.3) Classes d'utilisateurs (6)

Description des profils utilisateurs (DPU)

- **responsable-sécurité** : *Ingénieur en génie civil et environnement, ayant une expérience en lutte contre les incendies et en secours en montagne. Habitué à utiliser des outils nouveaux et techniques. A besoin de prendre des décisions très rapides. Pour cela, il est prêt à se former. Groupe peu nombreux et très homogène.* -----
- **garde** : *BTS eau et forêt, spécialiste de la montagne, connaît extrêmement bien la faune et la flore. Très sportif et observateur. Il doit avoir de l'autorité. C'est un groupe assez nombreux et dont l'attitude vis à vis de l'informatique varie du rejet complet à un intérêt marqué à condition que ce soit simple. Il n'aime pas les formations longues.* -----
- **ingénieur agronome** :



Chapitre 3 : Méthode de conception 00 d'une IHM

3.1) Présentation de la méthode

➤ **3.2) Études des besoins pour la CHM**

1) Objectifs & sous activités

2) Identification des domaines fonctionnels

3.2.3) Identification des classes d'utilisateurs

4) Analyse des tâches et de l'activité

5) Spécifications des performances d'usage

3.6.6) Synthèse

3.3) Spécifications de l'IHM

3.4) Réalisation et tests

3.2.4) Analyse des tâches et de l'activité (1)

- **Décrire le travail de l'utilisateur**
(2/3 des erreurs de conception sont commises à ce niveau)
- **Activité qui existe en dehors de la CHM**
- **Plusieurs méthodes (notations) plus ou moins complètes**
 - HTA (Hierarchical Task Analysis)
 - MUSE*/JSD
 - MAD : méthode analytique de description de tâche
 - GOMS
 - **Langage de Description de la Tâches Utilisateurs (LDTU)**

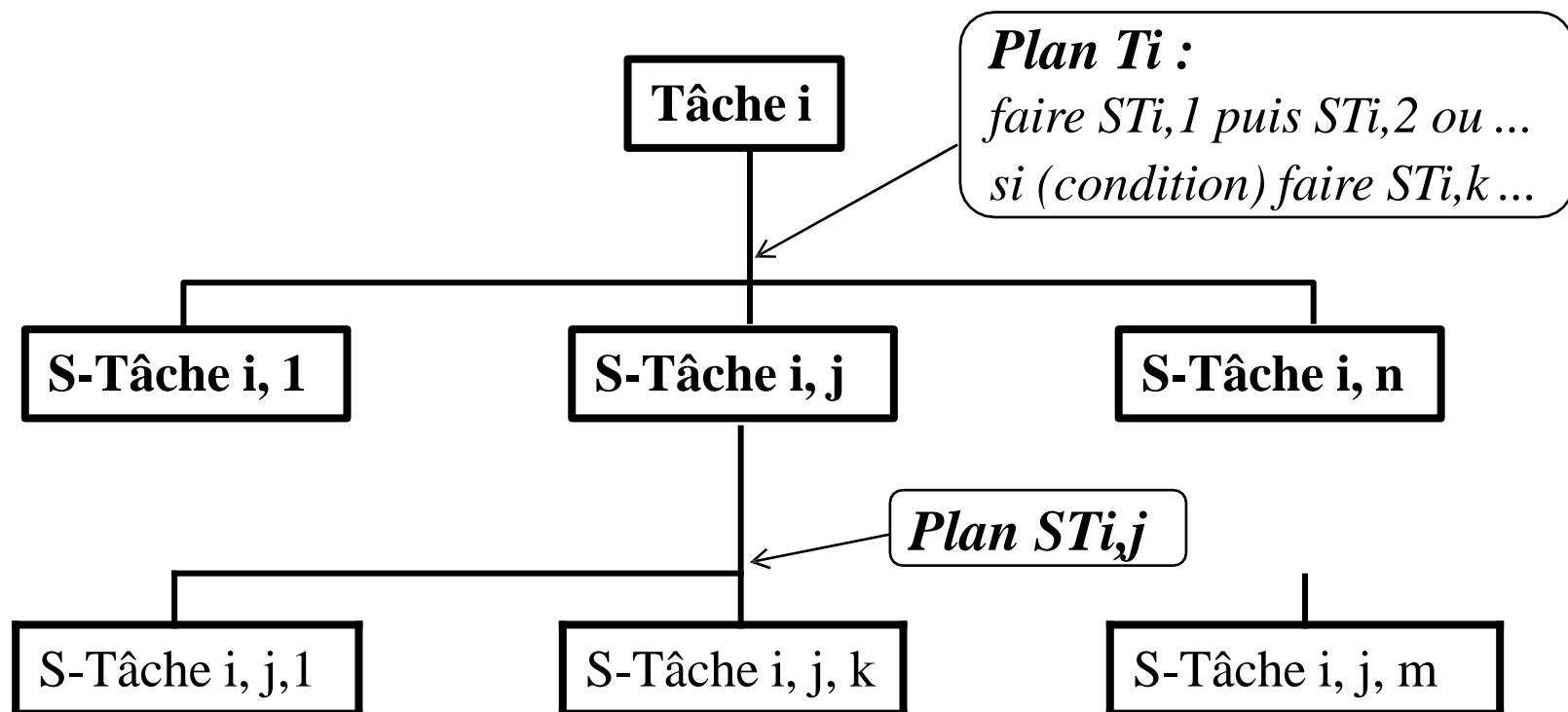
3.2.4) Analyse des tâches et de l'activité (2)



Méthodes d'analyse et de modélisation de la tâche humaine (suite)

HTA (Hierarchical Task Analysis)

- décomposition en tâches, sous-tâches, sous-sous-tâches
- plan d'exécution des tâches (optionnels)



A VOUS XXX



Expliquez le niveau B
Complétez le niveau C



A

Ti) qualif UE = IHM

Plan Ti :
faire Ti.1 puis Ti.2
si (échec à Ti.2) faire Ti,3

B

Ti.1) suivre enseignements

Ti.2) Faire DS

Ti.3) rattraper

Plan Ti.1 : -----

C

Ti.1.1) suivre cours

Ti.1.2) suivre TD

Ti.1.3)

3.2.4) Analyse des tâches et de l'activité (3)



➤ Méthodes d'analyse et de modélisation de la tâche humaine (*suite*)

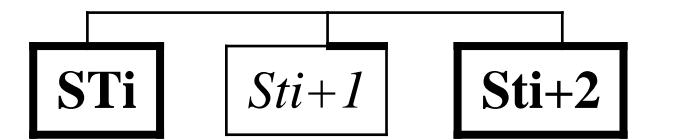
MUSE*/JSD

- arborescence des tâches et sous tâches
- distinction entre tâche humaine et automatisée

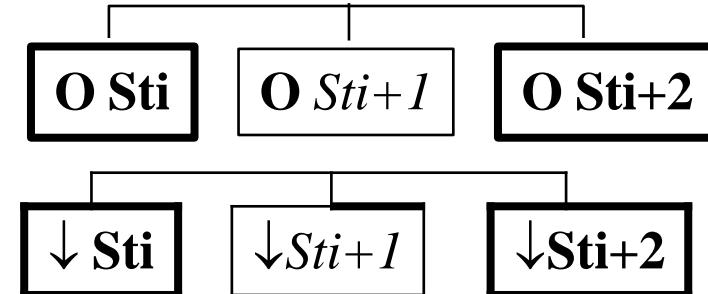


- distinction entre tâches séquentielles

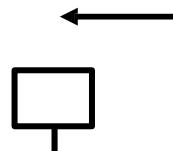
alternatives



et sans ordre



- possibilité de boucles



caractérisation

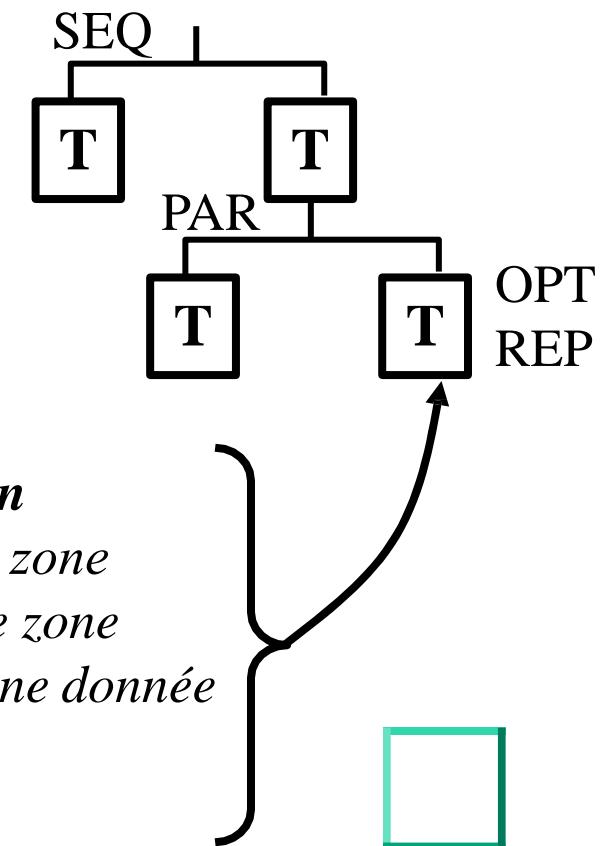
3.2.4) Analyse des tâches et de l'activité (4)



□ Méthodes d'analyse et de modélisation de la tâche humaine (suite)

➤ MAD : méthode analytique de description de tâche

- arborescence des tâches et sous tâches
- structuration des tâches composées avec les opérateurs :
 - SEQ : tâches en séquence (*ordre*)
 - AGG : agrégat de tâche (*sans ordre*)
 - PAR : tâches parallèles
 - ALT : tâches alternatives (CHO : choix)
 - OPT : tâche optionnelle (FAC : facultatif)
 - REP : tâches répétitive (BOUCLE)
- description de chaque tâche
 - Nom *contrôler altitude d'un avion*
 - État initial *altitude A lors de l'entrée en zone*
 - État final *altitude B lors de la sortie de zone*
 - But *optimiser le trafic sur une ligne donnée*
 - Précondition *vue radar*
 - Postcondition *hors zone de contrôle*



3.2.4) Analyse des tâches et de l'activité (5)



□ Méthodes d'analyse et de modélisation de la tâche humaine (*suite*)

➤ Autres méthodes

❖ GOMS :

- **Goal** : but de la tâche
- **Operator** : actions provoquant un changement d'état de l'utilisateur et de l'environnement
- **Method** : méthodes procédurales pour atteindre le but
- **Selection** : règle de sélection des sous-tâches
(si cond alors sous-tâche)

❖ Description des tâches avec des réseaux de Petri

3.2.4) Analyse des tâches et de l'activité (6)

□ Langage de Description de la tâches utilisateurs (LDTU)

- Approches linguistiques
 - modèle de tâche (ex une DTD ou un schéma XML)
 - instances de tâches (objets XML)
- Fondée sur l'approche algorithmique avec l'ajout d'un minimum de formalisme
- S'inscrit dans la méthode de conception d'IHM proposée

➤ distinction entre but (tâche) et action

- tâche \Leftrightarrow but
- sous-tâche \Leftrightarrow sous-but

} **OU alternatives**

- actions élémentaires
- actions composées

} **ET, OU, formes algorithmiques**
(si-alors-sinon, répéter, ...)

: Cf cours :
BD Avancées



3.2.4) Analyse des tâches et de l'activité (7)

❖ LDTU (*suite*)

- mettre en évidence les tâches en identifiant
 - les (sous) buts poursuivis \Leftrightarrow (sous) tâches
 - les actions à mettre en œuvre (*au niveau macroscopique*)
- *Point de départ : la table des types d'utilisateur par domaine fonctionnel (TTU/DF)*
- *Deux (sous) tâches doivent être réalisées indépendamment l'une de l'autre*
- *Une (sous) tâche élémentaire est décrite comme une succession d'action (optionnelle, répétitive, etc).*

3.2.4) Analyse des tâches et de l'activité (8)

➤ LDTU (*suite*)

- rappel des conventions pour décrire un langage

connecteurs

,	ET ordonné (<i>séquence</i>)
&	ET non ordonné (<i>agrégat</i>)
	OU exclusif (<i>choix</i>)

indicateurs d'occurrences

+	1 ou plusieurs fois
?	0 ou 1 fois
*	0,1 ou plusieurs fois

{..} groupe de symboles

‘ **en gras et rouge** ’ : les éléments du vocabulaire terminal;

3.2.4) Analyse des tâches et de l'activité (9)

LDTU (suite)

- *tâche* := id_tâche, titre, type?, *description_tâche* ;
- *id_tâche* := ‘T.’, identificateur hiérarchisé, ‘)’ ;
 - *titre* := texte ;
- *type* := [manuelle] | [système];
- *description_tâche* := f-algorithmique*, sous-tâches?;
- *sous-tâches* := (tâche+ | action+);
- *action* := id_action, f-condition?, titre, type?, *description_action* ;
 - *id_action* := ‘A.’, identificateur hiérarchisé, ‘)’ ;
- *f-condition* := ‘si (’, condition, ‘):’;
- *description_action* := f-algorithmique*, sous-action?;
- *sous-action* := (action+ | action-élémentaire+);
- *action-élémentaire* := id_action, f-condition?, titre, type?,
- *f-algorithmique* := f-séquence | f_agrégat | f_répétition | f_alternative | f-parallèle | f_optionnelle

3.2.4) Analyse des tâches et de l'activité (10)

❖ LDTU (*suite*)

➤ Grammaire du LDTU

- f-séquence := ‘, SEQ :’ (*option par défaut au niveau des actions*)
- f-alternative := ‘, ALT :’ (*option par défaut au niveau des tâches*)
- f-agrégat := ‘, AGG :’
- f-parallèle := ‘, PAR :’
- f-optionnelle := ‘, OPT :’
- f-répétitive := ‘, REP :’

-
- *Cette grammaire n'est pas normalisée ni complète. En fait, s'agit d'une description en langue quasi-naturelle avec un certain nombre de conventions au niveau du vocabulaire et de la présentation. Vous devez utiliser la présentation et les parenthèses pour rendre votre description lisible. Vous pouvez aussi adapter ces conventions.*
 - *Vous pouvez aussi utiliser une des méthodes ancienne (HTA, MUSE, MAD, GOMS).*

A VOUS X

Expliquez la tâche suivante

T.i) valider l'UE « IHM » : SEQ

 A.i.1) suivre enseignements : SEQ

 A.i.1.1) suivre le cours [manuelle]

 A.i.1.2) suivre les TD [manuelle]

 A.i.1.3) faire les TP [système]

 2) faire le DS [manuelle]

 3) si (échec) : passer le rattrapage [manuelle]

A VOUS X

décrivez la tâche à l'aide du LDTU qui permet à un élève d'obtenir **a qualification pour le domaine « Développement logiciel » : valider les UE « USDP », « IHM », « Qualité » (parallélisme possible, pas d'ordre imposé)**

3.2.4) Analyse des tâches et de l'activité (11)

❖ Exemple

k) tâche de l'hôtesse d'accueil (T.H.A) : ALT

T.k.1) T.H.A. dans la gestion de la sécurité

3) T.H.A. dans la gestion des visiteurs

T.k.3.1) répondre au téléphone [*manuelle*]

2) gérer l'agendas des visites [*système*]

3) accueillir les visiteurs : SEQ

A.k.3.1.1) faire fiche visiteur [*système*]

2) appeler le guide concerné [*manuelle*]

3) si absent : -----

3.2.4) Analyse des tâches et de l'activité (12)

- **la planification hiérarchique de la tâche :**

- **hiérarchisation de la tâche**

- ✓ racine : tâche initiale \Leftrightarrow *but initial*
 - ✓ nœuds : tâches et sous tâches \Leftrightarrow *buts & sous-buts*

} *transactions*

Attention à la granularité

- **elle permet**

- d'imaginer la manière dont l'utilisateur percevra la tâche
 - de définir les contraintes temporelles
 - de détecter des éléments communs à plusieurs tâches

- **on cherche à équilibrer**

- la hauteur de l'arbre
 - la largeur de l'arbre

3.2.4) Analyse des tâches et de l'activité (13)

❖ Le Livrable : PHTU-d

Planification **Hiérarchique** de la **Tâche Utilisateur**

Construire **TTU/DF** : **Table de la Tâche Utilisateur par Domaine Fonctionnel**

- ligne : 1 profil utilisateur (pu)
 - colonne : 1 domaine fonctionnel (df) } intersection
TU/DF(pu, df)
-
- identifier les **Tâches Utilisateur par Domaine Fonctionnel (TU/DF)** non vides
 - **ITU/DF** : Identificateur de la **Tâches d'un Utilisateur dans un Domaine Fonctionnel**

T.numero-utilisateur.numero-domaine-fonctionnel

ex: **T.k.3** : *Tâche de l'hôtesse d'accueil (Uk) dans le DF "administration des visiteurs (D3)*

- décrire chaque TU/DF avec le LDTU
- DTU/DF** : **Description** de la **Tâches d'un Utilisateur dans un Domaine Fonctionnel**

3.2.4) Analyse des tâches et de l'activité (14)

<i>TTU/DF : Table de la Tâche Utilisateur par Domaine Fonctionnel</i>	D1) Sécurité	D2) Admin. du parc	D3) Admin visiteurs	D4) Com.	Dj)	Dn) admin système
U1) Externe				<i>T.1. i</i>		
U2) Visiteur	<i>T.2.1</i>		<i>T.2.3</i>	<i>T.2.4</i>		
Ui) Forestier	<i>T.i.1</i>	T.i.2	<i>T.i.3</i>	<i>T.i.4</i>		
Uj) Garde	<i>T.j.1</i>	T.j.2	<i>T.j.3</i>	<i>T.j.4</i>		
Uk) Hôtesse	<i>T.k.1</i>		T.k.3	T.k.4		<i>T.k.n</i>
Up) Ing. Agro	<i>T.p.1</i>	T.p.2	<i>T.p.3</i>	<i>T.p.4</i>		
Uq) Vétérinaire	<i>T.q.1</i>	T.q.2		<i>T.q.4</i>		
Ux) Resp. Sécu parc	T.x.1	<i>T.x.2</i>	<i>T.x.3</i>	T.x.4		<i>T.x.n</i>

Uz) Resp. SI	<i>T.z.1</i>	<i>T.z.2</i>	T.z.3	<i>T.z.4</i>		T.z.n

TU/DF(k,3) : Tâche de "Hôtesse " dans de DF "Administration des visiteurs"

T.k.3 : Identificateur Tâche de "Hôtesse " dans de DF "Administration des visiteurs"

Evaluation de l'implication : **T.u.d : important** | **T.u.d : moyen** | **T.u.d : faible**

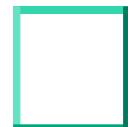
3.2.4) Analyse des tâches et de l'activité (15)

Formalisme des DTU/DF (*Description de TU/DF*)

Identificateur DTU/DF	évent. id. du DTU/DF père	Profil Utilisateur	Domaine Fonctionnel
Description des tâches réalisées dans le DTU/DF avec le langage LDTU			lien vers DAU ou DTU/DF fils

DAU : **Description des Actions Utilisateurs** - cf spécifications fonctionnelles

- **Dans l'étude des besoins**, on se limite au niveau des tâches (buts, sous-but, sous-sous-buts, etc.) : il s'agit donc du début de l'analyse de la tâche. Celle-ci sera affinée dans les spécifications fonctionnelles
 - il n'y a pas d'actions (uniquement des tâches, sous-tâches, etc.);
 - la seule forme algorithmique est l'alternative (ALT)
 - à ce niveau, il n'est pas obligatoire de distinguer les tâches
 - ❖ purement manuelles
 - ❖ toujours en relation avec le système
 - ❖ mixtes





3.2.4) Analyse des tâches et de l'activité (16)

- Comment lire TTU/DF ?
- Début de l'arbre de la planification hiérarchique de la tâche

Tâche Utilisateur dans la gestion du parc: ALT

T.1) Tâche de « Externe » :

T.1.4) Tâche de « Externe » dans le DF "Communication"

T.2) Tâche du "Visiteur" : ALT

T.2.1) Tâche du "Visiteur" dans le DF "Sécurité"

T.2.3) Tâche du "Visiteur" dans le DF "Administration des visiteurs"

T.2.4) Tâche du "Visiteur" dans le DF "Communication"

T.k) Tâche de "l'hôtesse d'accueil" : ALT

T.k.1) Tâche de "l'hôtesse " dans le DF "Sécurité"

T.k.3) Tâche de "l'hôtesse " dans le DF "Administration des visiteurs"

T.k.4) Tâche du de "l'hôtesse " dans le DF "Communication"

T.k.n) Tâche de "l'hôtesse " dans le DF "Administration Système"

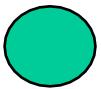
T.z) Tâche du "Responsable SI" : ALT

T.z.1) Tâche du " Responsable SI " dans le DF "Sécurité"

T.z.n) Tâche du " Responsable SI " dans le DF " Administration Système "

Redondant avec TTU/DF

3.2.4) Analyse des tâches et de l'activité (17)



➤ Décrire chaque TU/DF DTU/DF (Description de TU/DF)

DTU/DF N° 1		U1 : Externe	DF3 : administration des visiteurs
--------------------	--	---------------------	---

ALT:

T.1.3.1) consulter le site internet

T.1.3.2)

DTU/DF N° 2		U2 : Visiteur	DF1 : Sécurité
--------------------	--	----------------------	-----------------------

ALT :

T.2.1.1) signaler un danger

T.2.1.2)

DTU/DF N° 3		U2 : Visiteur	DF3 : administration des visiteurs
--------------------	--	----------------------	---

ALT :

T.2.3.1) consulter le site internet (*idem T.1.3.1*)

T.2.3.2)

3.2.4) Analyse des tâches et de l'activité (18)

DTU/DF N° n

Up : ingénieur agronome

DF1 : gestion de la sécurité

ALT

- 1) signaler un danger (*idem T.2.1.1*)
- 2)

DTU/DF N° n+1

Up : ingénieur agronome

DF2 : administration du parc

ALT

- 1) gérer la flore : ALT
 - T.p.2.1.1) gérer les plantations
 - T.p.2.1.2) gérer les traitements
 - T.p.2.1.3)
 - 2) gérer la faune : ALT
 - 1) gérer les naissances
 - 2) gérer les vaccinations
 - T.p.2.2.3) gérer les maladies
 - T.p.2.2.4)
 - T.p.2.3) gérer les parcelles
-

X
à vous de compléter

3.2.4) Analyse des tâches et de l'activité (19)

DTU/DF N° m

Uk : hôtesse d'accueil

DF1 : gestion de la sécurité

ALT

- 1) signaler un danger (*idem T.2.1.1*)
 - 2) gérer les fiches d'accident
 - 3) gestion des communications avec les secours
- T.k.1.4) -----

DTU/DF N° m+1

Uk : hôtesse d'accueil

DF3 : admin. des visiteurs

ALT :

- T.k.3.1) répondre au téléphone [*manuelle*]
 - T.k.3.2) gérer l'agendas des visites [*système*]
 - T.k.3.3) accueillir les visiteurs
 - 4) gérer les statistiques
 - 5) gérer la caisse visiteur
- T.k.3.6) -----

à vous de compléter

3.2.4) Analyse des tâches et de l'activité (20)

DTU/DF N° y

Ux : responsable sécurité

DF1 : gestion de la sécurité

*à vous de
compléter*

DTU/DF N° y+1

Ux : responsable sécurité

DF2 : gestion du parc

*à vous de
compléter*

3.2.4) Analyse des tâches et de l'activité (21)

AU/DF N° y +2

UX : responsable sécurité

DF3 : admin. des visiteurs

ALT :

T.x.3.1) traiter un appel au secours

T.x.3.2) évaluation préventive des risques

T.x.3.3).....

❖ **Le Livrable : PHTUd (*début*) se compose :**

- **TTU/DF :** Table de la Tâche Utilisateur par Domaine Fonctionnel
- **les DTU/DF :** Description de Activité d'un Utilisateur dans un Domaine Fonctionnel (**description des cases non vides de TITU/DF:**

Chapitre 3 : Méthode de conception 00 d'une IHM

3.1) Présentation de la méthode

➤ **3.2) Études des besoins pour la CHM**

- 1) *Objectifs & sous activités*
- 2) *Identification des domaines fonctionnels*
- 3.2.3) *Identification des classes d'utilisateurs*
- 3.2.4) *Analyse des tâches et de l'activité*

✓ **3.2.5) Spécifications des performances d'usage**

- 3.6.6) *Synthèse*
- 3.3) *Spécifications de l'IHM*
- 3.4) *Réalisation et tests*

3.2.5) Spécifications des performances d'usage (1)

➤ Les performances d'usage

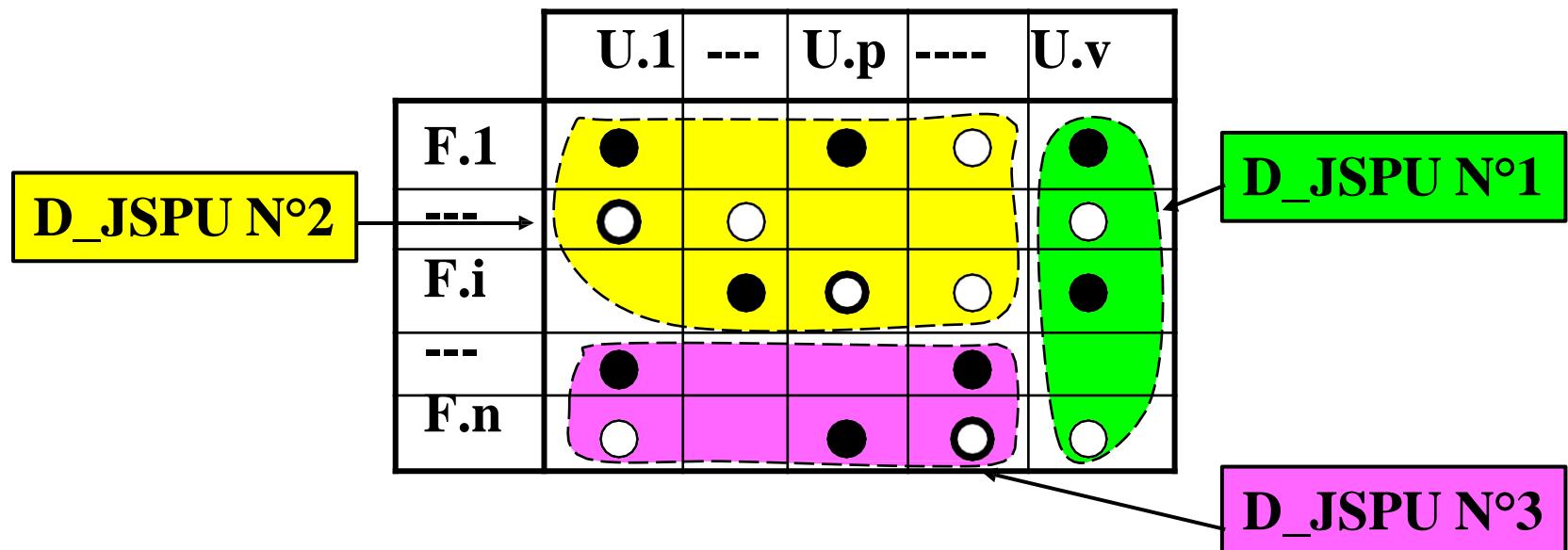
- représentent **les objectifs à atteindre** dans la réalisation de l'interface, indépendamment des choix de conception;
 - définissent qualitativement et quantitativement **l'utilisation** du système par la population cible;
 - elles prennent en compte aussi :
 - ❖ la compréhension des messages et symboles
 - ❖ le caractère prédictible des réponses du système
 - ❖ la récupérations en cas d'erreur humaine
 - les exigences en termes de : ✓ complexité des opérations
✓ temps de réponse
✓ coût des erreurs
- *prévoir des moyens pour contrôler*

3.2.5) Spécifications des performances d'usage (2)

❖ Définir les jeux de spécifications de performances d'usage (JSPU).

On définit un JSPU pour un ensemble cohérent de TU/DF : *le domaine du JSPU (D_JSPU)*

- Pour les applications critiques : Un JSPU est fait pour chaque TU/DF (fonction/utilisateur). *Exemple : pilotage d'une centrale nucléaire*)
- Pour les applications très simple : un seul JSPU est réalisé pour l'ensemble de l'application (*site internet grand public ou un jeu*)
- Pour les applications intermédiaires : Quelques JSPU (entre 3 et 10) sont faits par groupe TU/DF (*application de gestion classique*)



3.2.5) Spécifications des performances d'usage (3)

❖ Définir pour chaque domaine les spécifications de performance d'usage

- objectifs à atteindre
- utilisation du système
- compréhension des messages et symboles
- le caractère prédictible des réponses
- la récupération en cas d'erreurs
- la complexité des opérations
- le temps de réponse
- le coût des erreurs
- etc

**Jeu de
SPU**

3.2.5) Spécifications des performances d'usage (4)

Exemple : Les D_JSPU pour l'application Parc du Mont AGALAC

The legend indicates four types of JSPU (Juste Saisie Pour Utilisation) represented by different patterns:

- D_JSPU N°1: Green horizontal lines
- D_JSPU N°2: Blue wavy lines
- D_JSPU N°3: Pink vertical lines
- D_JSPU N°4: Red grid

	D1) Sécurité	D2) Admin. du parc	D3) Admin visiteurs	D4) Com.	Dj) -----	Dn) admin système
U1) Externe				T.1.i		
U2) Visiteur	T.2.1		T.2.3	T.2.4		
Ui) Forestier	T.i.1	T.i.2	T.i.3	T.i.4		
Uj) Garde	T.j.1	T.j.2	T.j.3	T.j.4		
Uk) Hôtesse	T.k.1		T.k.3	T.k.4		T.k.n
Up) Ing. Agro	T.p.1	T.p.2		T.p.4		
Uq) Vétérinaire	T.q.1	T.q.2		T.q.4		
Ux) Resp. Sécu parc	T.x.1	T.x.2	T.x.3	T.x.4		T.x.n

Uz) Resp. SI	T.z.1	T.z.2	T.z.3	T.z.4		T.z.n

4.3) Besoins

4.3.5) performances d'usage (5)

❖ *Exemple : JSPU pour le domaine N° 4 (le plus critique)*

- ✓ **SPU(4,1) : objectifs à atteindre** : Traitement immédiat de tout appel. Rester en contact avec la personne qui a lancé l'appel jusqu'à l'arrivée des secours. Guider les sauveteurs. Contrôler l'opération jusqu'à sa conclusion. Limiter les dégâts induits. Définir des priorités claires. -----
- ✓ **SPU(4,2) : utilisation du système** : Tout employé du parc doit être capable de prendre en compte l'appel et de déclencher la procédure d'urgence sans avoir reçu aucune formation. Celle-ci est ensuite conduite par un responsable de la sécurité qui a reçu une formation spécifique. ---
- ✓ **SPU(4,3) : compréhension des messages et symboles** : Les messages doivent être significatifs et brefs. Ils doivent pouvoir être émis en plusieurs langues (au choix de l'utilisateur). Utiliser des mots simples et précis. Les symboles à utiliser doivent reprendre les signalisations graphiques internationales. -----

4.3) Besoins

4.3.5) performances d'usage (6)

- ✓ **SPU(4,4) : le caractère prédictible des réponses** : les réponses doivent être prédictibles dans la mesure du possible. -----
- ✓ **SPU(4,5) : la récupération en cas d'erreurs** : Enregistrer toutes les opérations effectuées avec le maximum d'informations (ordre, heure, qui, ...). Pouvoir faire, défaire et refaire sans limitations. Déetecter si possible les incohérences de l'opérateur et les signaler pour son remplacement éventuel. -----

- ✓ **SPU(4,6) : complexité des opérations** : Succession d'opérations simples même dans le cas où la situation devient complexe (n suites d'opérations simples et guidées si nécessaires). -----
- ✓ **SPU(4,7) : temps de réponse** : immédiat (quelques secondes), arrêter si nécessaire les traitements d'importance secondaires qui pourraient freiner.
- ✓ **SPU(4,8) : coût des erreurs** : coût d'une ou plusieurs vies humaines (très élevé). -----

Chapitre 3 : Méthode de conception 00 d'une IHM

3.1) Présentation de la méthode

➤ **3.2) Études des besoins pour la CHM**

3.2.1) Objectifs & sous activités

3.2.2) Identification des domaines fonctionnels

3.2.3) Identification des classes d'utilisateurs

3.2.4) Analyse des tâches et de l'activité

3.2.5) Spécifications des performances d'usage

□ 3.2.6) Synthèse

3.3) Spécifications de l'IHM

3.4) Réalisation et tests

3.2.6) Synthèse (1)

➤ **Synthèse** : montrer que les besoins ont été bien identifiés

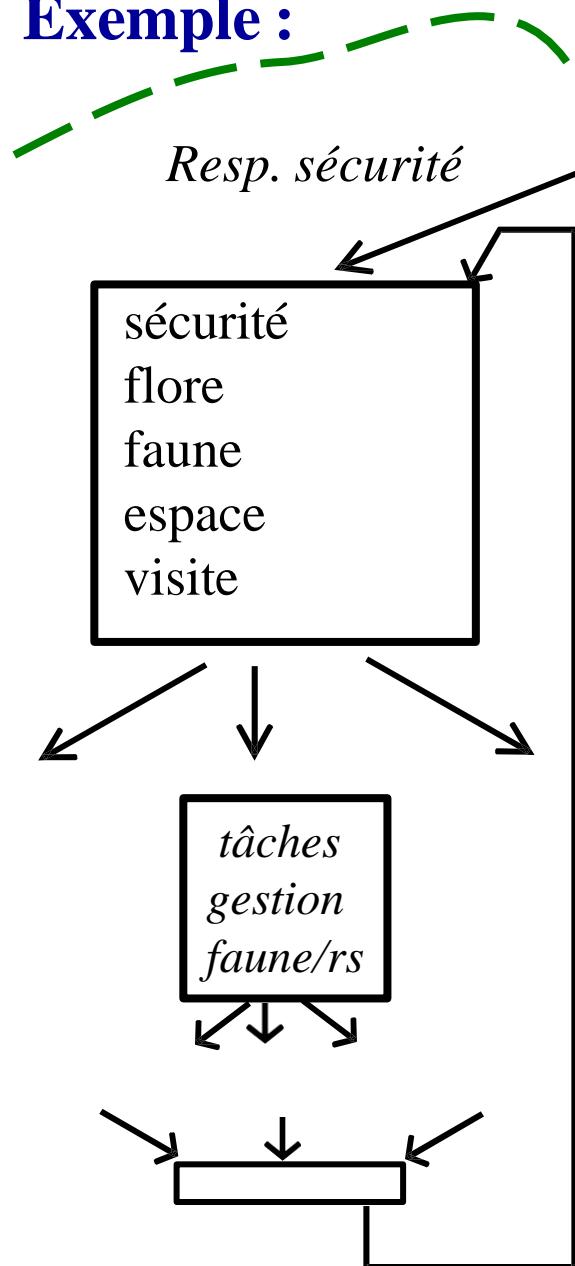
- domaines fonctionnels
- profils d'utilisateurs
- principales tâches
- performances d'usage

➤ **souvent une maquette jetable** (*choix ergonomiques non faits*)

- maquette papier
- maquette électronique (*outil de présentation ?*)
 - enchaînement des fenêtres
 - explication des tâches (texte, schéma, ...)

3.2.6) synthèse (2)

Exemple :



Maquette de capture des besoins (2)

3.2.6) synthèse (3)

□ L'étude des besoins pour la CHM est une tâche difficile

- qui doit s'appuyer en particulier sur des interviews des futurs utilisateurs:
 - ❖ Combien de personnes à interroger ?
 - ❖ Comment les choisir ?
 - ❖ Que leur demander ?
- qui doit être vérifiée et validée:
 - ❖ prévoir des procédures de vérification et de validation

3.2.6) synthèse (4)

Liste des livrables de l'étude des besoins (IHM)

DTU : Description des Tâches de l'Utilisateur

DDF Description des domaines fonctionnels ★

MU Modèle des utilisateurs

GPU Graphe des profils utilisateurs

DPU Description des profils d'utilisateurs ★

PHTU_d Planification hiérarchique des tâches des utilisateurs (*début*) ★
TTU/DF et les DTU/DF

SPU Spécifications des performances d'usage ★

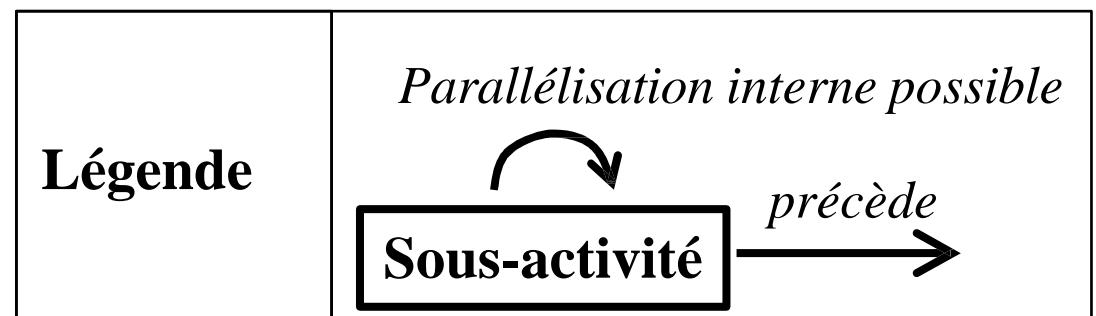
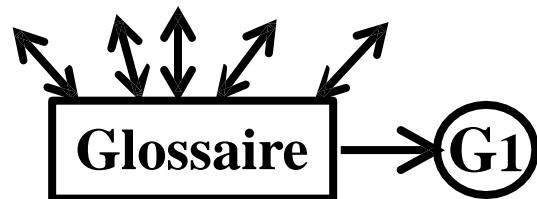
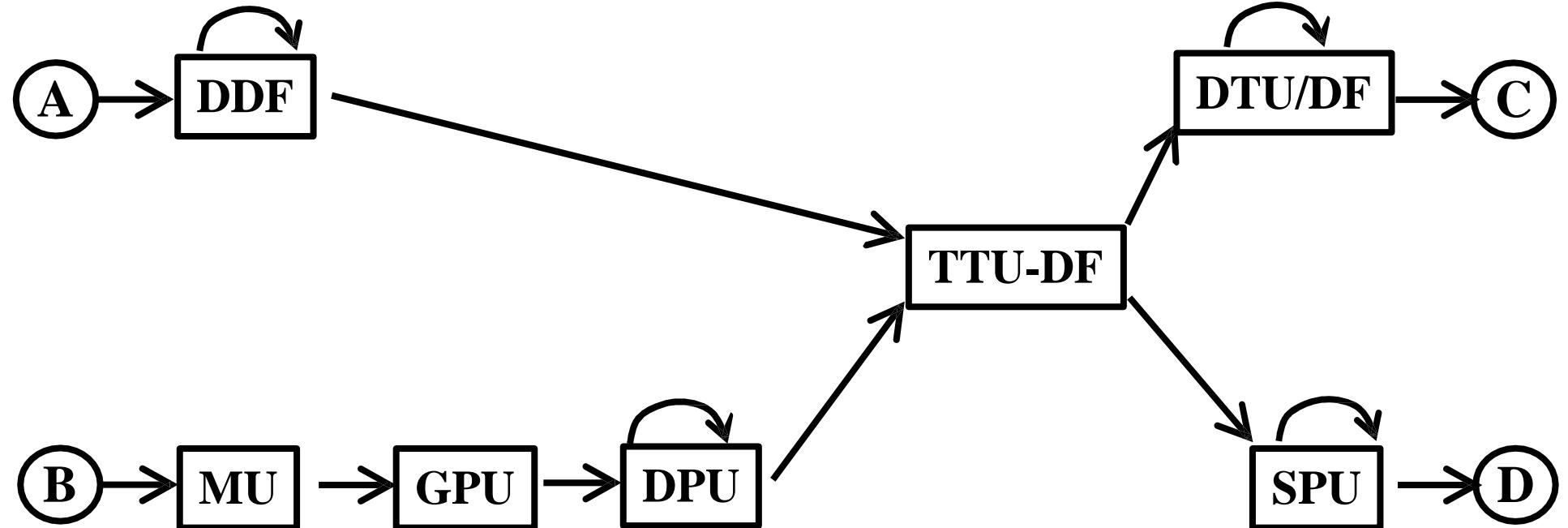
+ **GLOS** glossaire ★

+ **MAQ-CB** maquette de capture des besoins

★ : A faire dans les petites applications

3.2.6) synthèse (5)

➤ Ordonnancement des sous-activités de l'étude des besoins :



Légende

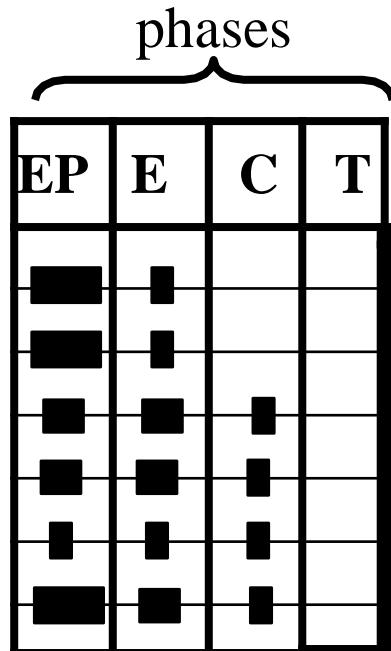
Parallélisation interne possible

Sous-activité

précède

4.3.6) synthèse (6)

➤ Etude des besoins pour la CHM dans les phases USDP



- analyse des domaines fonctionnels
- analyse des classes d'utilisateurs (*profils*)
- analyse des tâches et de l'activité
- spécifications des performances d'usage
- mettre à jour le glossaire
- réaliser une maquette de capture des besoins

DTU : *Description des Tâches utilisateur*

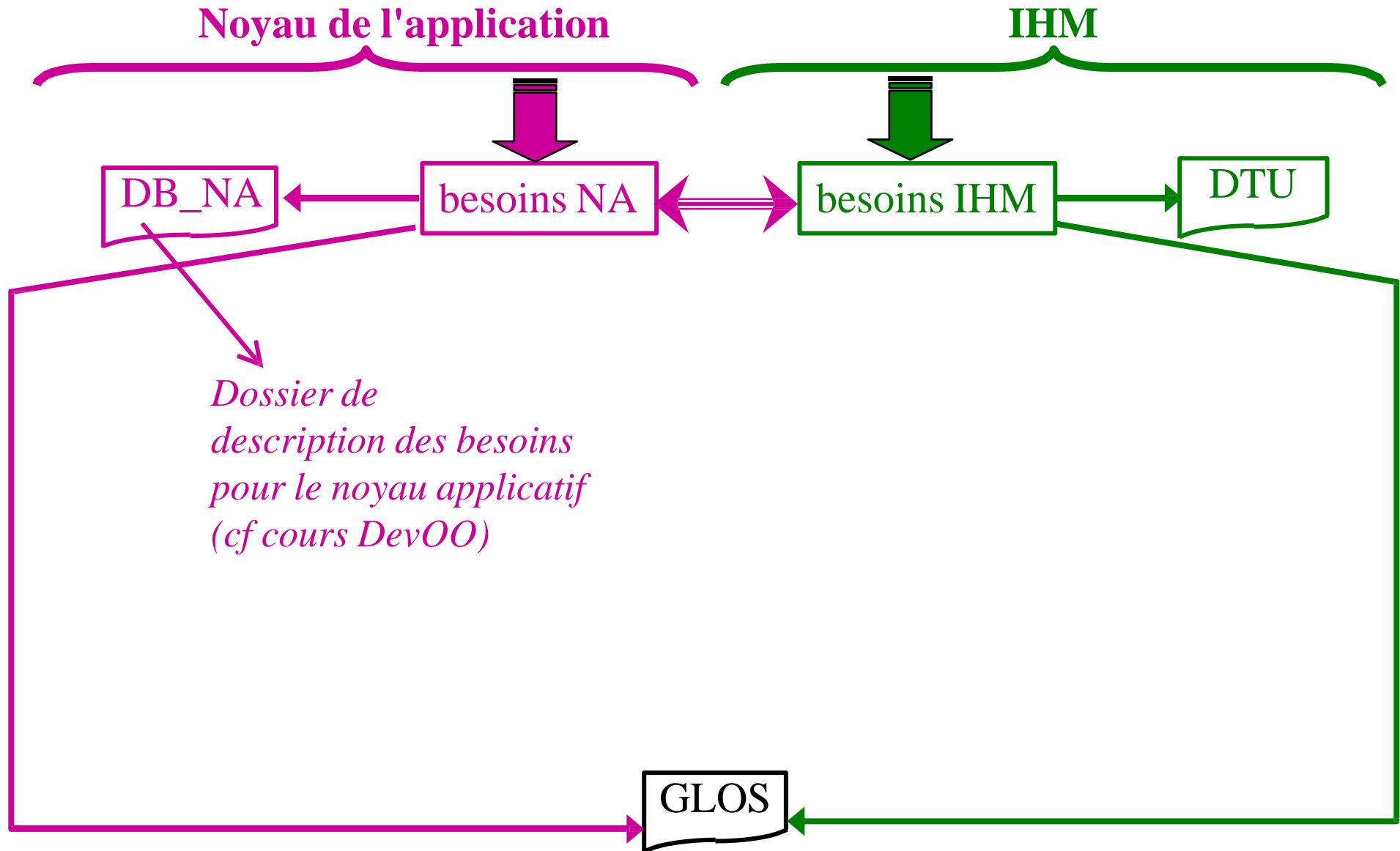
EP: étude préliminaire

E : élaboration

C : Construction

T : transition

2.3.6) synthèse (7)



Chapitre 3 : Méthode de conception 00 d'une IHM

- 1) *Présentation de la méthode*
- 2) *Études des besoins pour la CHM*

➤ 3.3) Spécifications de l'IHM

□ 3.3.1) introduction

- 2) spécifications conceptuelles
- 3) spécifications fonctionnelles
- 4) spécifications syntaxiques
- 5) spécifications lexicales

3.4) Réalisation et tests

Un sondage au hasard

- ❖ Quelles relations existe-t-il entre l'étude du noyau de l'application et l'étude de l'IHM ?
- ❖ Rappelez les 4 sous activités de l'étude des besoins !
- ❖ Qu'est ce qu'un profil utilisateur ?

3.3.1) Introduction pour les spécifications de l'IHM (1)

➤ Objectifs

- prendre en compte à la fois :
 - les aspects ergonomiques
 - l'ensemble des problèmes techniques de mise en œuvre
- Cette méthode
 - aborde la conception d'une IHM en terme **de langage d'interaction**
 - tente de prendre en compte tous les types d'interfaces

➤ 4 Étapes

- ✓ spécifications **conceptuelles**) → domaine du signifié
- ✓ spécifications **fonctionnelles**) → domaine du signifiant
- ✓ spécifications **syntaxiques**) → domaine du signifiant
- ✓ spécifications **lexicales**) → domaine du signifiant

3.3.1) Introduction pour les spécifications de l'IHM (2)

➤ Spécification conceptuelle (*Niveau but et tâches*)

✓ A partir de la planification hiérarchique définir :

- les objets manipulés dans chaque tâche
- leurs attributs
- leurs relations
- les opérations que l'on peut réaliser sur ces objets

Que
peut-on
faire ?

➤ Spécification fonctionnelle (*Niveau sémantique*)

✓ définir (au niveau signification)

- l'effet des actions de l'utilisateur sur les objets
- la signification des retours du système

Que se
passe-t-il
si ?

3.3.1) Introduction pour les spécifications de l'IHM (3)

❖ Spécification syntaxique (*détail du dialogue*)

- ✓ les séquences d'E/S qui représentent au niveau
 - :• de la tâche : les procédures les objectifs
 - du système : les règles d'utilisation

Comment
faire pour
?

❖ Spécification lexicale (*représentation φ et codage*)

- ✓ sur le plan **de la tâche** prendre en compte :
 - la terminologie
 - l'image mentale } des utilisateurs
- ✓ sur le plan **du système** viser
 - : une bonne compréhension
 - une bonne mémorisation } des symboles

Que signifie telle
information ?

❖ *pour ces 2 derniers niveaux, on se réfère généralement à un guide de style*

Chapitre 3 : Méthode de conception 00 d'une IHM

- 1) *Présentation de la méthode*
- 2) *Études des besoins pour la CHM*

➤ 3.3) Spécifications de l'IHM

3.3.1) *introduction*

✓ 3.3.2) spécifications conceptuelles

- 3) spécifications fonctionnelles
- 4) spécifications syntaxiques
- 5) spécifications lexicales

3.4) Réalisation et tests

3.3.2) Spécifications conceptuelles (1)

□ Objectif -> modèle conceptuel de l'utilisateur

MCU = Image (MCC)
métaphore ?

✓ MCC : le modèle conceptuel du concepteur (MCC)

NA

✓ MCU: le modèle conceptuel de l'utilisateur (MCU)

IHM

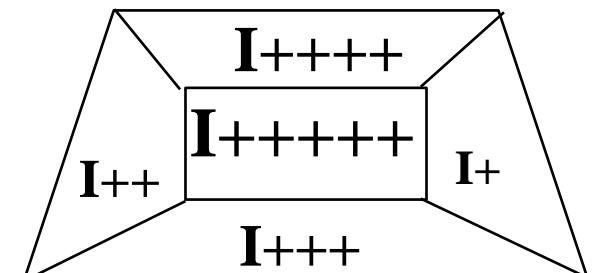
Métaphore :

- Vocabulaire
- Symboles (graphiques, sonores)
- Règles d'agencement, ...

3.3.2) Spécifications conceptuelles (2)

exemple d'une métaphore : automobile

- ❖ vocabulaire : volant, moteur, chauffeur, vitesse, freiner
- ❖ symboles graphiques :
 - ✓ couleurs : rouge = interdit, vert = autorisé, orange = possible
 - ✓ forme  = interdiction  = danger  = autorisation
- ❖ symboles sonores
 - ❖ aigu et fort = "urgence"
 - grave et faible = "prudence"
- ❖ règle d'agencement
 - ✓ organisation de tableau de bord de voiture
 - au centre en gros, le plus important
 - importance décroissante :
le haut, le bas, la gauche, la droite
 - ✓ etc.



3.3.2) Spécifications conceptuelles (3)

□ 1) Initialiser la conception de l'IHM : choix de la (des) métaphore(s)

- ❖ fondées sur les connaissances antérieures de l'utilisateur
 - ✓ une métaphore par profil utilisateur (*ou une pour plusieurs profils*)
- ❖ qualités : cohérence - généralité - extensibilité
- ❖ types de métaphores
 - **métaphores fonctionnelles** : interactions de haut niveau (objectif)
 - *exemple : conduire Mr X de A à B en 10mn max en respectant le code la route*
 - **métaphores opérationnelles** : interactions de bas niveau : liste d'opérations; *exemple :*
 - début*
 - choisir la voiture V
 - faire monter Mr X
 - entre A et A+a accélérer pour passer la vitesse de 0 à 50 km/h
 - rouler de A+a et B- b rouler à vitesse constante de 50 km/h
 - entre B-b et B ralentir pour passer la vitesse de 50 à 0 km/h
 - faire descendre Mr X

fin

3.3.2) Spécifications conceptuelles (4)

□ 1) Initialiser la conception de l'IHM (suite)

❖ choisir un guide de style standard (*Windows, Linux, Web, etc.*)
ou commencer à le définir

□ *guide de style :*

- *document de référence de l'entreprise (souvent volumineux et structuré) qui doit être appliqué par tous ;*
- *il contient un ensemble de règles qui :*
 - *sont fondées sur la métaphore*
 - *doivent être appliquées dans la présentation et la structuration du dialogue : icônes, noms, symboles, couleurs, polices, sons, formes, structure des interactions, disposition, langue(s), etc*
 - *contribuent à la cohérence et à la facilité d'utilisation de l'application*
- *rédigé généralement par une équipe multi-métiers (ergonomes, informaticien, psychologue, sociologue, etc.)*

3.3.2) Spécifications conceptuelles (5)

□ 1) Initialiser la conception de l'IHM (suite)

- ❖ **Adapter le guide de style à la charte graphique de l'organisation**
- **Charte graphique:**
 - *document de référence de l'entreprise qui doit être appliqué par tous ;*
 - *il contient un ensemble consignes pour la communication :*
 - ✓ *logo de l'organisation*
 - ✓ *couleurs et police de caractères recommandées*
 - ✓ *modèles physiques de documents*
 - *rédigé généralement par une équipe multi-métiers (ergonomes, psychologue, sociologue, etc.)*
 - *pas nécessairement lié à l'informatique (par exemple la signalétique des bâtiments, des rues, etc.*

3.3.2) Spécifications conceptuelles (6)

□ 1) Initialiser la conception de l'IHM (suite)

➤ exemple "extrait de la charte graphique de l'INSA-LYON"

- logocouleur :
 - Les carrés : les départements
 - les ronds : les laboratoires
 - etc.
- Interdiction de superposer un logo d'une entité propre à l'INSA (*IF ne peut pas avoir de logo*)
- le logo doit figurer :
 - à gauche pour un département (le nom du département est à droite)
 - à droite pour un laboratoire (le nom du laboratoire est à gauche)
 - au centre pour service administratif (le nom du service est au centre sous le logo)
 - lorsque une entité dépend de plusieurs établissements, le logo de ces établissements peut côtoyer celui de l'INSA (même taille).
exemple le LIRIS dépend de l'INSA, de l'UCB et du CNRS.



A VOUS X

- Vous êtes chargé de développer une application pour permettre le "vote électronique" en Algérie
 - ❖ quelle type de métaphore choisissez-vous pour le grand public ?

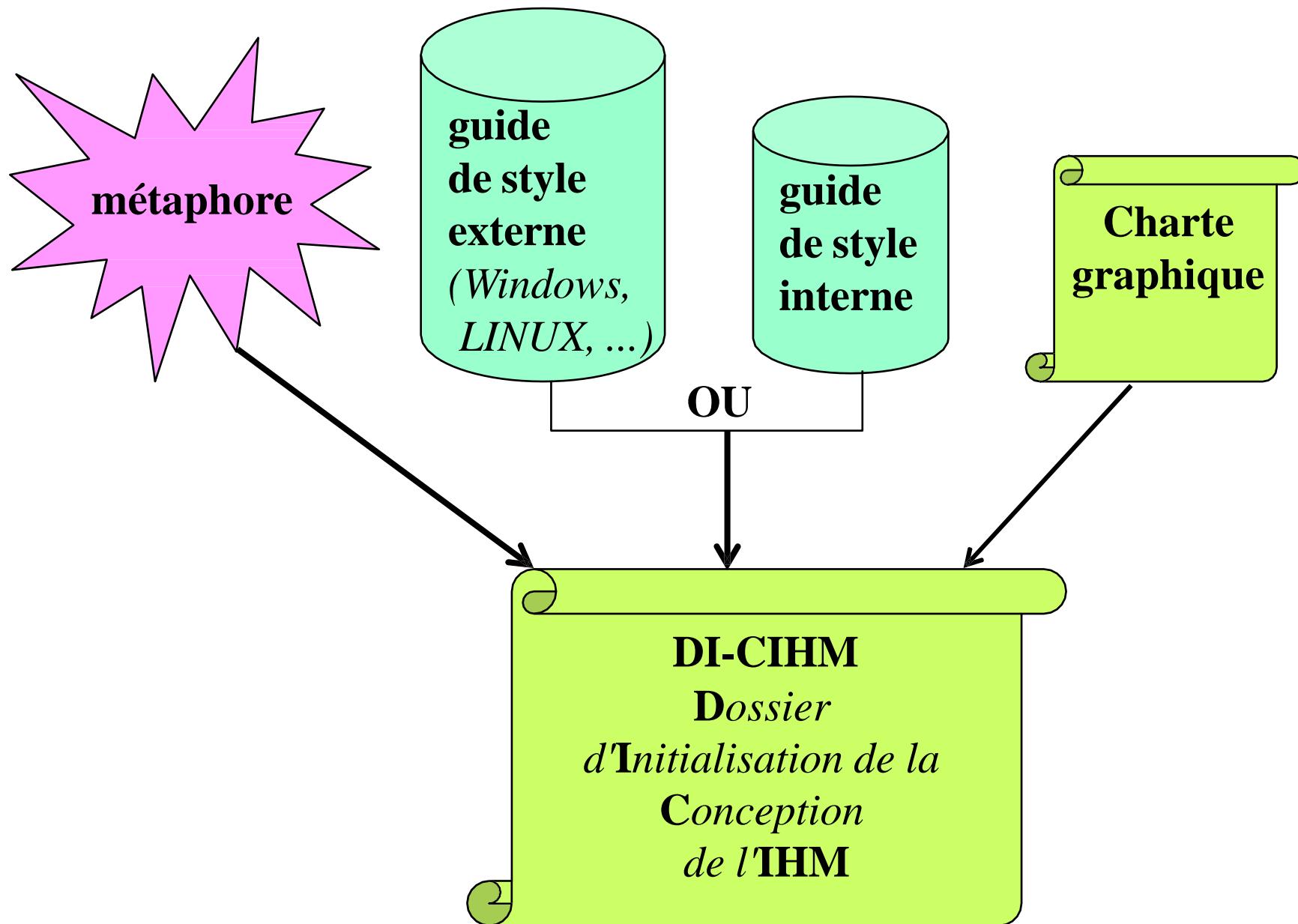
A VOUS X

- ❖ Donnez une esquisse du guide de style que vous proposeriez

A VOUS X

- ❖ Comment constitueriez-vous l'équipe chargée de rédiger le guide de style ?

4.4.2) Spécifications conceptuelles (7)



3.3.2) Spécifications conceptuelles (8)

□ 1) Initialiser la conception de l'IHM (fin)

❖ Contenu de DICIHM : Dossier d'Initialisation de la Conception de l'IHM

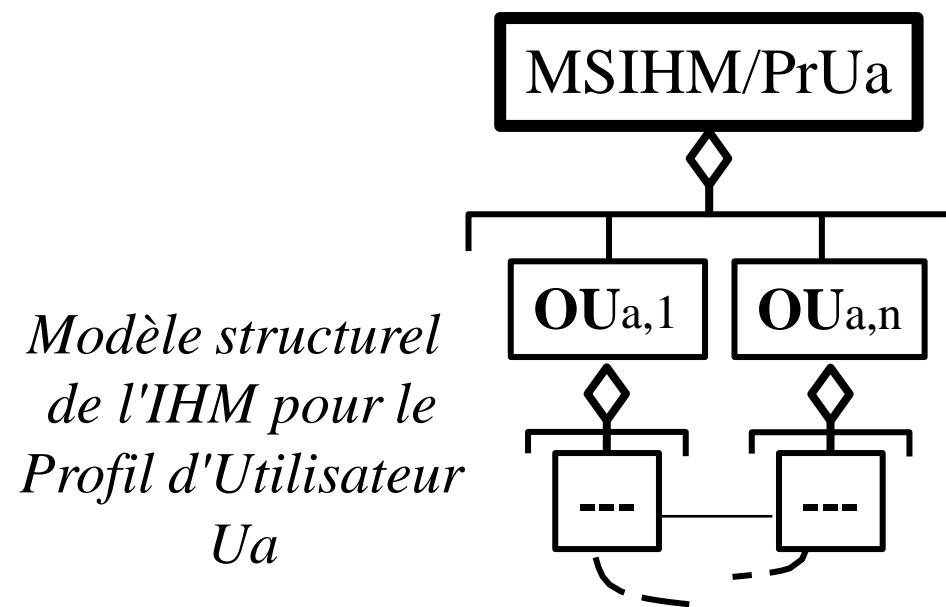
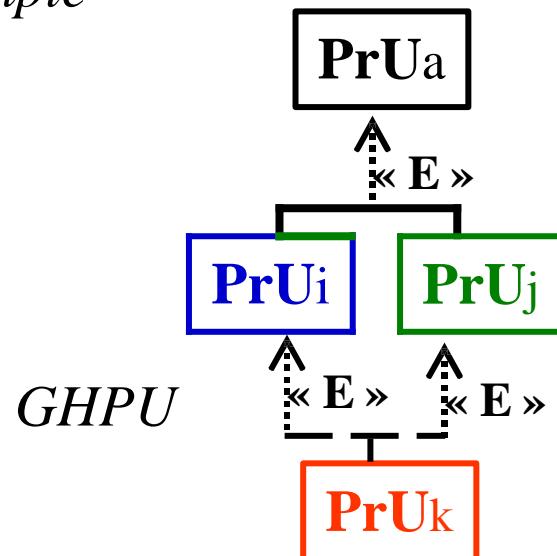
- rappel de la charte graphique de l'établissement (optionnel)
- choix d'un guide de style : ALT
 - reprise d'un guide standard (WINDOWS, LINUX,)
 - utilisation du guide de style de l'établissement (SNCF, EDF, ...)
 - guide de style spécifique à l'application (commencer à le décrire)
- pour chaque profil utilisateur (ou groupe de profils): le type et l'identification de la métaphore utilisée
 - description textuelle de la métaphore (quelques lignes)
 - liste des **Principaux Objets Utilisateurs (POU)**
s'appuyer sur : le début de la Planification Hiérarchique des tâches des utilisateurs (PHTUd)

3.3.2) Spécifications conceptuelles (9)

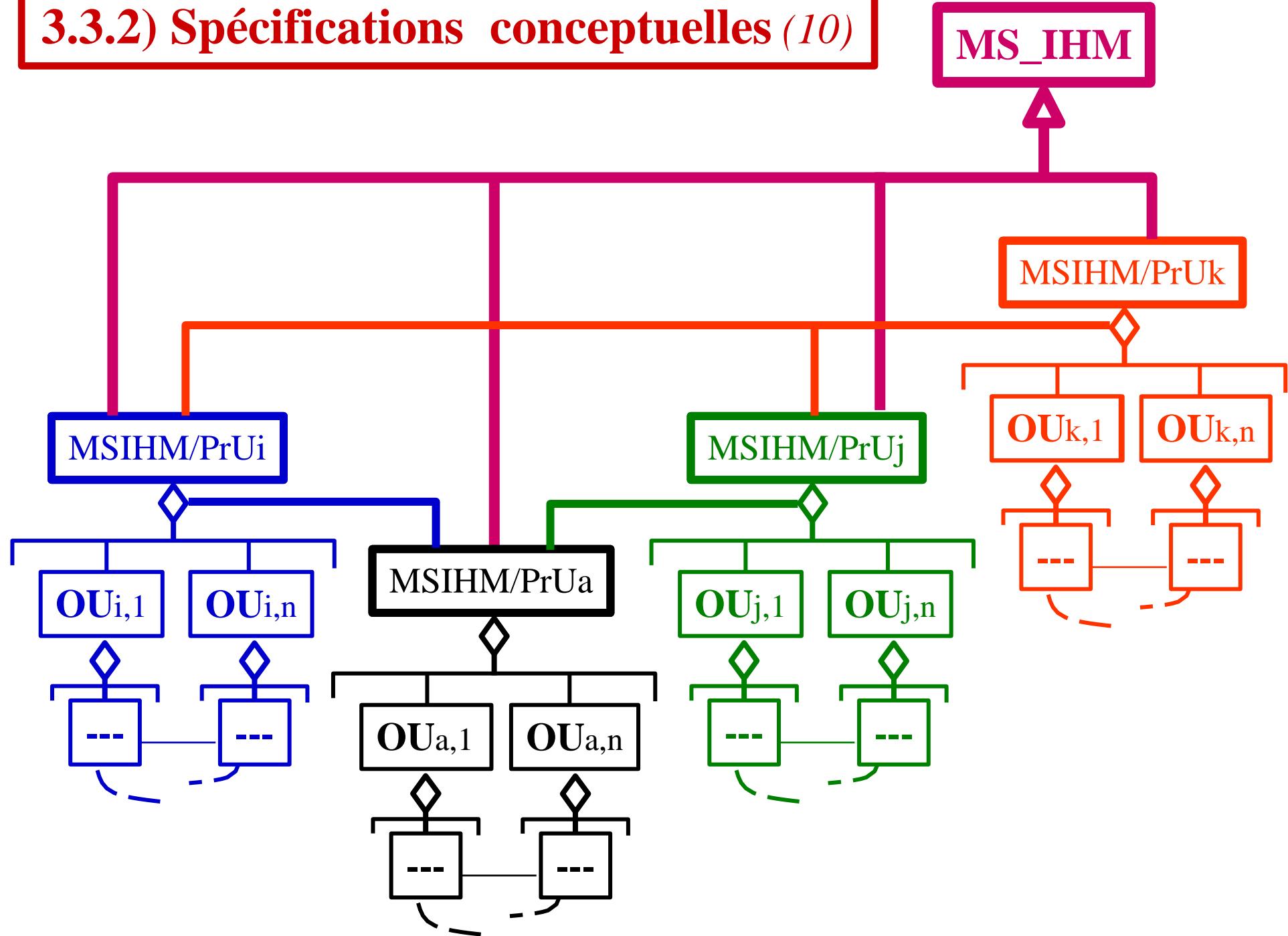
□ 2) Construire le modèle structurel de l'IHM (MS_IHM)

- identifier les relations entre les POU
- sélectionner et structurer les POU
- Construire le modèle structurel de l'IHM (au sens UML et USDP)
 - ✓ une spécialisation de l'IHM par profil utilisateur
 - ✓ commencer par le plus simple
 - ✓ tenir compte des extensions de profils pour construire les autres spécialisations

exemple



3.3.2) Spécifications conceptuelles (10)



3.3.2) Spécifications conceptuelles (11)

□ Description des POU (DPOU)

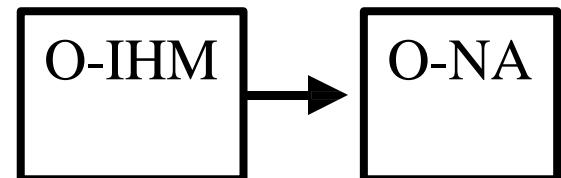
- **donner une description détaillée**
 - de ces objets -> propriétés visuelles (présentation)
-> propriétés non visuelles (description)
 - de ces actions que peut réaliser l'utilisateur

□ Liaison avec le modèle structurel de l'application (MSApp)

cf USDP

->liens entre objets de l'IHM et objets du noyau de l'application

- *O-IHM connaît O-NA*
- *O-NA ne connaît pas O-IHM*



□ Compléter le glossaire (USDP)

3.3.2) Spécifications conceptuelles (12)

Exemple

Exemple : Dossier d'Initialisation de la Conception de l'IHM

- choix des métaphores
 - tous sauf pour les administrateurs du SI : métaphore fonctionnelle
 - administrateurs du SI : métaphore fonctionnelle et métaphore opérationnelle (*cet IHM ne sera pas développé dans cet exemple*).
- Le parc dispose d'une charte graphique qui doit être respectée (cf doc ccccc - *non développé dans cet exemple*)
- Le parc dispose **d'un** guide de style du parc (cf doc ggggg *non développé dans cet exemple*)
- Pour chaque profil (garde, vétérinaire, ingénieur agronome, hôtesse d'accueil,) reprise du **vocabulaire des métiers correspondants** (métiers de la montagne et du loisir --- à détailler). On utilisera des graphiques réaliste pour les objets naturels (animaux, végétaux, humains, montagne, ...)

3.3.2) Spécifications conceptuelles (13)

Exemple

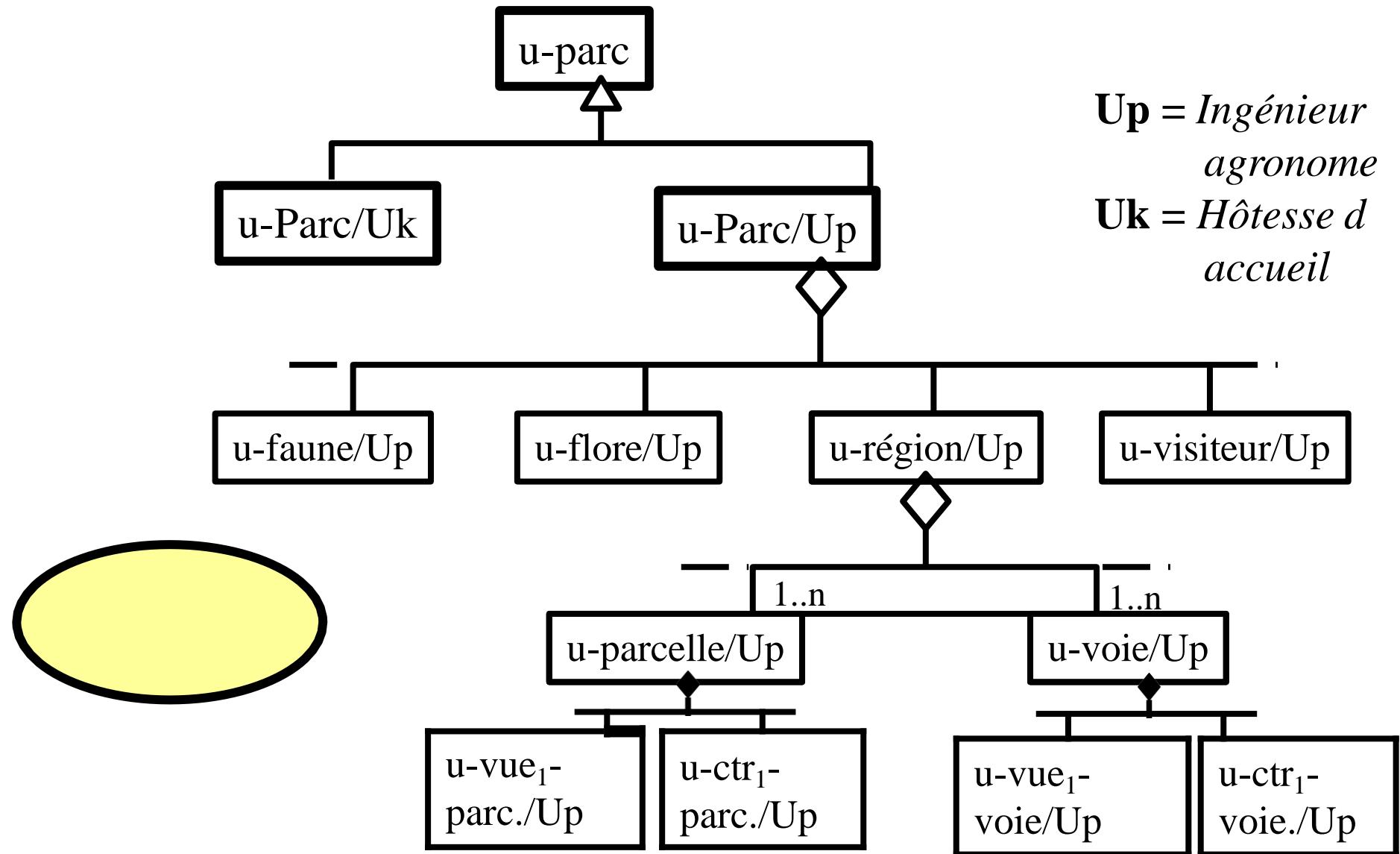
Exemple : Dossier d'Initialisation de la Conception de l'IHM

POU	par profil	Uk	U.	Uq	Up	Ux
		hôtesse	xxx	vétérin.	Ing Agr	resp séc
u-parc				X	X	X
u-faune				X	X	X
u-flore					X	X
u-forêt				X	X	
u-région					X	
u-parcelle					X	
u-animal				X		
u-voie				X		
u-visite						
U-vaccin	X			X		
-----	<i>à vous de compléter</i>					
u-sos						X

3.3.2) Spécifications conceptuelles (14)

Exemple (suite)

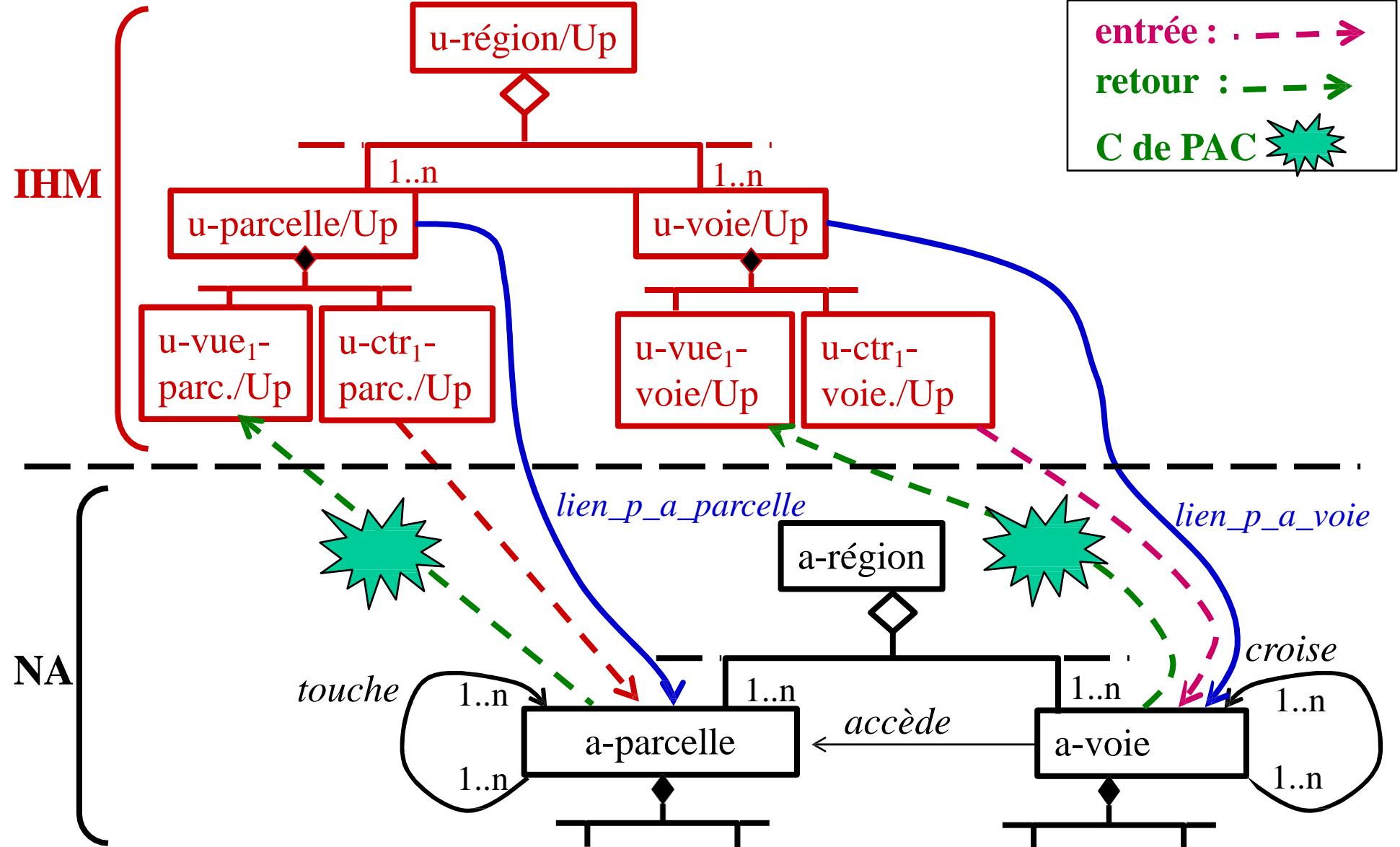
Modèle structurel de l'IHM (MSIHM)



3.3.2) Spécifications conceptuelles (15)

Exemple (suite)

❖ Modèle structurel de l'IHM (MSIHM) – liaison avec MS_NA



3.3.2) Spécifications conceptuelles (16)

Exemple (*suite*)

u-parc

□ Description des principaux objets DPOU

définition	Objet unique représentant le parc naturel qui regroupe tous les composants (racine du modèle)
présentations	<ol style="list-style-type: none">1) Icône (le logo du parc) qui permet de lancer l'application <i>(accessible à tous les types d'utilisateurs)</i>2) Tableau : adresse, téléphone, fax, mel, adresse http, <i>(accessible à tous les types d'utilisateurs)</i>3) Description textuelle du parc <i>(destiné à tout public, accessible par tel portable)</i>4) Série de photographies et vidéo <i>(destiné à tout public, accessible par INTERNET)</i>5) -----
actions	<ol style="list-style-type: none">1) consulter les présentations 2, 3, 4, 5 (<i>tous</i>)2) gérer les présentations 2, 3, 4, 5 (<i>resp. relation externe</i>)3) lancer l'application (<i>tous, le résultat dépend du type de l'utilisateur</i>)4) -----

3.3.2) Spécifications conceptuelles (17)

Exemple (*suite*)

- *description des principaux objets DPOU (suite)*

u-animal

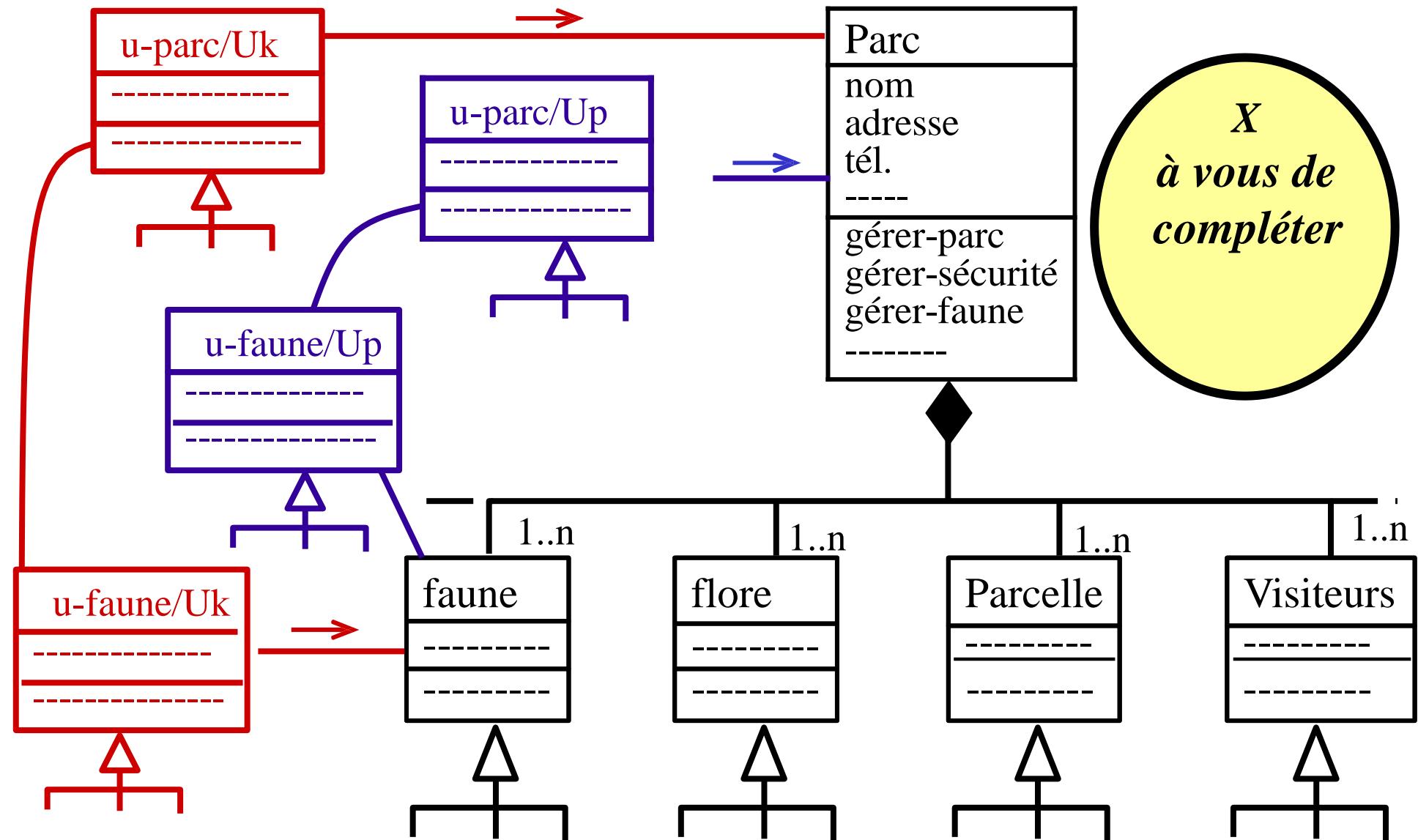
définition	Individu vivant dans le parc, suffisamment intéressant pour être identifié (l'ourson 'Bouba')
présentations	<ol style="list-style-type: none">1) Tableau donnant les caractéristiques de cet individu : famille, groupe, âge, parents, localisation dans le parc, ----- <i>(accessible à tous sauf externe)</i>2) Série de photographies de cet individu <i>(accessible à tous sauf externe)</i>3) Informations complémentaires sur sa santé <i>(accessible aux vétérinaires et aux gardes)</i>4) -----
actions	<ol style="list-style-type: none">1) consulter les présentations 1, 2, <i>(tous sauf externe)</i>2) gérer les présentations 1, 2, sauf localisation <i>(technicien animalié)</i>3) gérer la présentation 3 <i>(vétérinaire)</i>4) mettre à jour la localisation <i>(garde)</i>5) -----

à vous de compléter

3.3.2) Spécifications conceptuelles (18)

Exemple (suite)

❖ Liaison avec le modèle structurel de l'application (MSApp)



3.3.2) Spécifications conceptuelles (19)

Exemple (*suite*)

- ❖ Compléter le glossaire de donnée (*cf USDP*)

Symbol	definition	type
parc	objet du NA représentant le parc (racine)	objet
animal	objet du NA représentant un animale	objet
-----	-----	----
id-fa	identificateur d'une famille animale (NA)	Oid
nom-fa	nom d'une famille animale (NA)	chaîne-car
-----	-----	----
gérer-parc	méthode permettant de gérer les attributs de 'parc'	méthode
u-parc/Uk	objet de l'IHM permettant aux utilisateur de profil Uk (hôtesse) d'accéder au système d'information du parc	objet
u-ani/Up	objet de l'IHM permettant aux utilisateur de profil Up 'Ing. agr.' d'accéder a la partie du SI du parc concernant les animaux	objet
-----	-----	----
u-voir-vidéo	méthode permettant de visualiser une vidéo (IHM)	méthode
-----	-----	----

à vous de
compléter

3.3.2) Spécifications conceptuelles (20)

Synthèse des spécification conceptuelles de l'IHM

DC-IHM : Description Conceptuelle de l'IHM

DICIHM Dossier d'initialisation de la conception de l'IHM ★

MS_IHM Modèle structurel de l'IHM

DPOU Description détaillée des objets
(définition, présentations, actions) ★

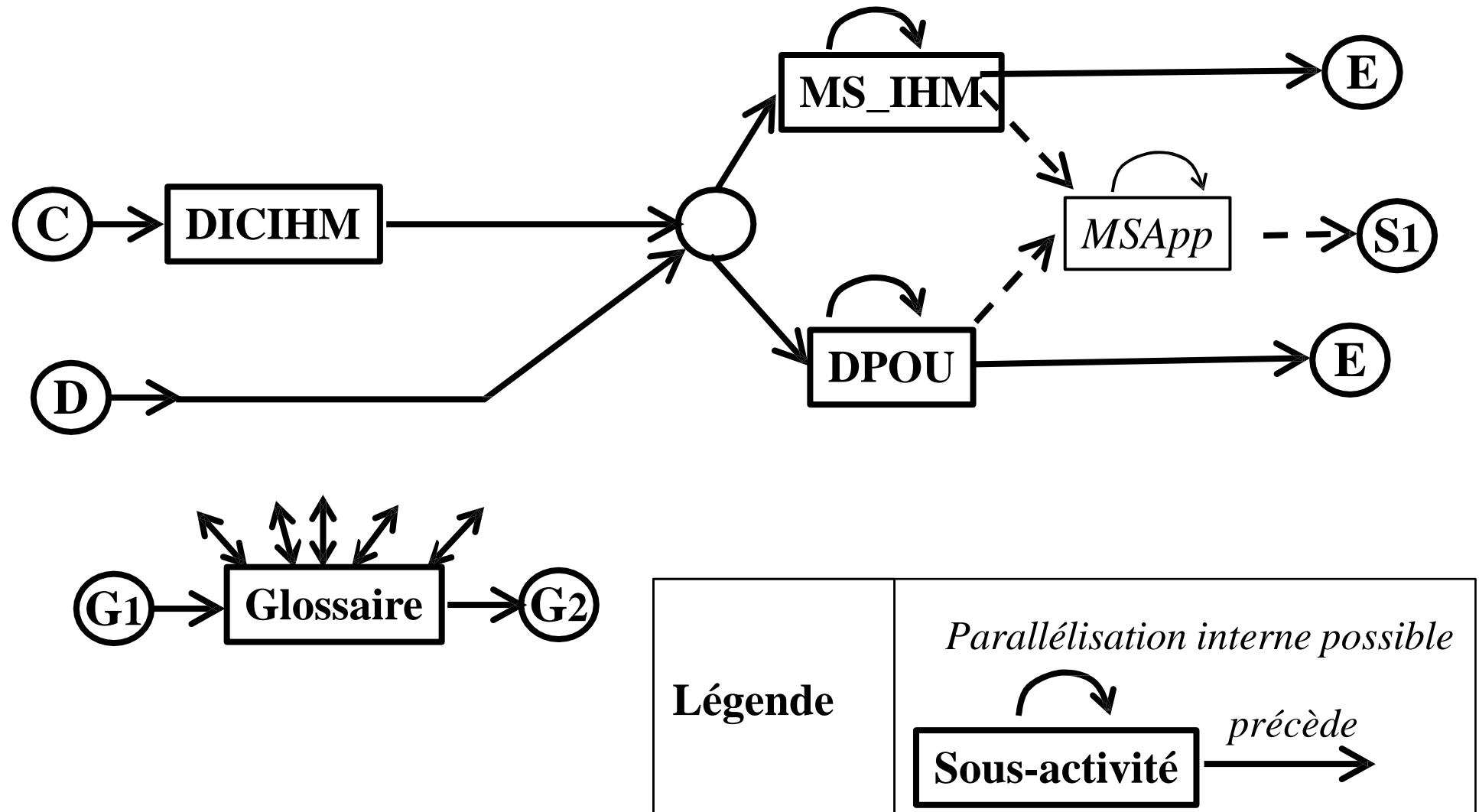
+ **MSApp** Liaison avec le modèle structurel de l'analyse ★

+ **GLOS** Mise à jour du glossaire ★

★ : Petites applications

3.3.2) Spécifications conceptuelles (21)

❖ Ordonnancement des sous-activités des spécifications conceptuelles

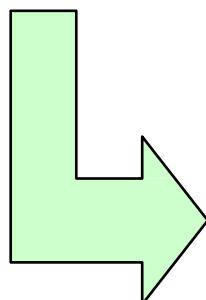


3.3.2) Spécifications conceptuelles (22)

Spécifications conceptuelles dans les phases USDP

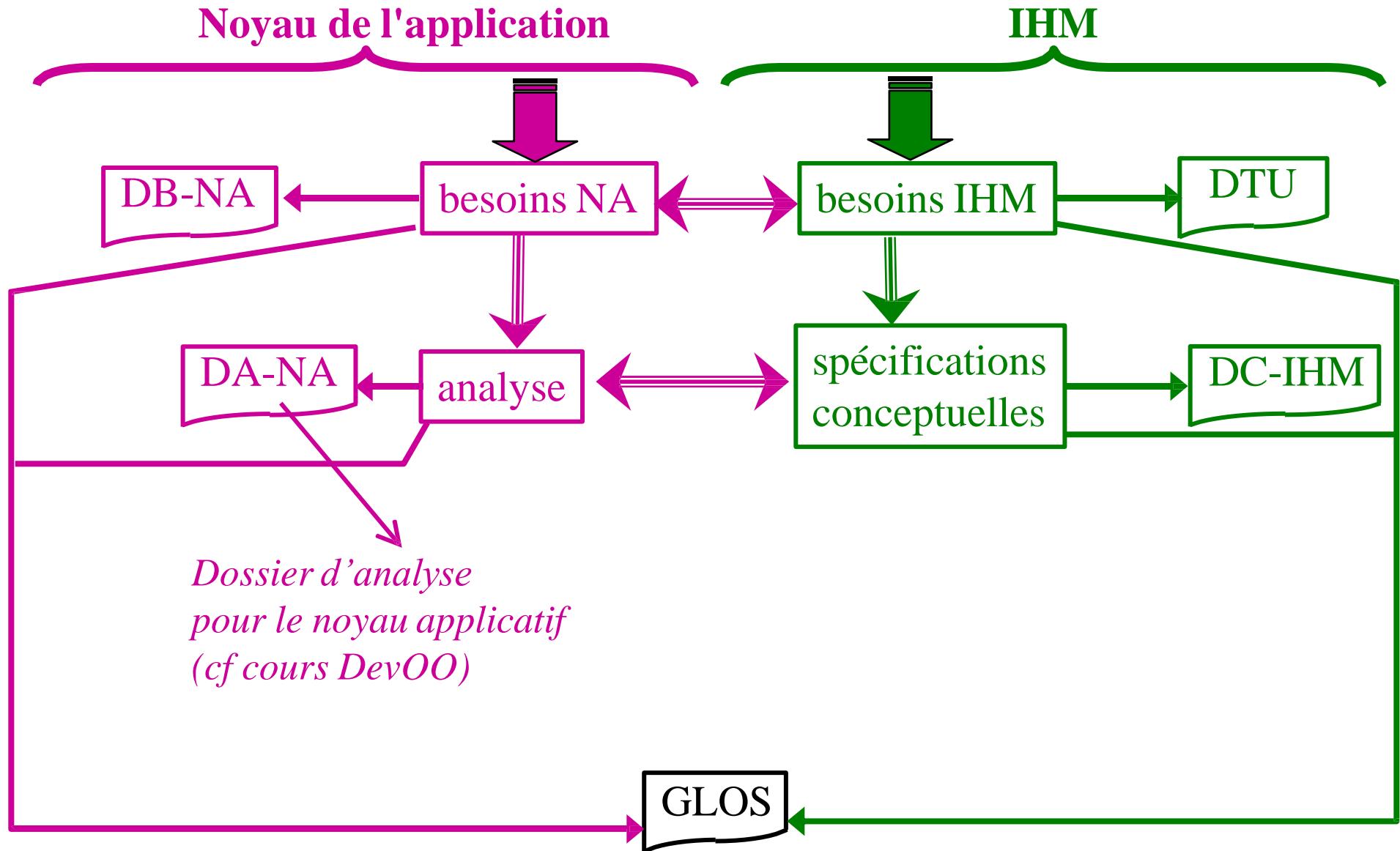
EP	E	C	T
■ ■ ■	■ ■ ■	■ ■ ■	— — —
■ ■ ■	■ ■ ■	■ ■ ■	— — —
■ ■ ■	■ ■ ■	■ ■ ■	— — —
■ ■ ■	■ ■ ■	■ ■ ■	— — —

- initialiser la conception de l'IHM
- description des POU
- construction du modèle structurel de l'IHM
- coupler avec le modèle structurel du noyau de l'application
- mettre à jour le glossaire



DC-IHM : *Description conceptuelle de l 'IHM*

3.3.2) Spécifications conceptuelles (23)



Chapitre 3 : Méthode de conception 00 d'une IHM

- 1) *Présentation de la méthode*
- 2) *Études des besoins pour la CHM*

➤ 3.3) Spécifications de l'IHM

- 1) *introduction*
- 2) *spécifications conceptuelles*

□ 3.3.3) spécifications fonctionnelles

- 4) spécifications syntaxiques
- 5) spécifications lexicales

3.4) Réalisation et tests

3.3.3) Spécifications fonctionnelles (1)

□ Sémantique de l'interface

Objectif : décrire l'effet des actions (*vue de l'utilisateur*)

-> effet de ces actions sur les objets (spécifications conceptuelles)

- elle permet à l'utilisateur de comprendre l'effet de ses actions
- elle est indépendante de la manière dont les actions seront effectivement réalisées (*interface graphique, textuelle, ...*).
- elle complète l'analyse de l'application (noyau de l'application)

3.3.3) Spécifications fonctionnelles (2)

1) Identifier les commandes (actions élémentaires)

- ❖ **Approfondir** la Planification Hiérarchique de la Tâche Utilisateur de commencée dans l'étude des besoins (PHTU-d) :

-> Planification Hiérarchique de la Tâche Utilisateur approfondie (**PHTU-a**)

- racine : tâche initiale \Leftrightarrow *but de l'application*

- nœuds : \rightarrow tâches et sous tâches \Leftrightarrow *buts & sous-buts*

->actions composées ou élémentaires } *A réaliser dans PHTU-a*
Attention à la granularité

- #### ▪ **Objectif** : actions /Profil Utilisateur

composées élémentaires

✓ *action élémentaire* \Leftrightarrow commandes

3.3.3) Spécifications fonctionnelles (3)

□ 1) Identifier les commandes (suite)

❖ Comment construire PHTU-a

- On repart
- des Description de la Tache Utilisateur par Domaine fonctionnel DTU/DF
- et on les complète avec
 - des **Description des Actions Utilisateur DAU**
 - des liens fils vers les DAU (identificateurs de DAU)

Exemple de DTU/DF complétée avec les liens fils

DTU/DF N° m+1	Uk : hôtesse d'accueil	DF3 : administration des visiteurs
ALT :	T.k.3.1) répondre au téléphone [<i>manuelle</i>] T.k.3.2) gérer l'agendas des visites [<i>système</i>] -----> T.k.3.3) accueillir les visiteurs -----> T.k.3.4) gérer les statistiques -----> T.k.3.5) gérer la caisse visiteur -----> T.k.3.6) -----	liens DAU (t) DAU (v) DAU (w) DAU (x) -----

3.3.3) Spécifications fonctionnelles (4)

exemple (suite)

✓ noeud père

DAU N° v	DTU/DF N° m+1	Uk : hôtesse d'accueil DF3 : administration des visiteurs
T.k.3.3) accueillir les visiteurs ALT :	a.k.3.3.1) gérer les visites non guidées -----> a.k.3.3.2) gérer les visites guidées -----> -----	liens DAU(v+1) DAU(v+2)
DAU N° v+1	DAU/DF N° v	Uk : hôtesse d'accueil DF3 : administration des visiteurs
T.k.3.3.1) gérer les visites non guidées SEQ :	a.k.3.3.1.1) enregistrer le visiteur [système] -----> a.k.3.3.1.2) enregistrer son programme [système] -----> a.k.3.3.1.3) délivrer son billet [système] -----> a.k.3.3.1.3) l'autoriser à commencer sa visite [système] -- ----- a.k.3.3.1.z) enregistre la fin de sa visite [système] ----->	liens C: enreg-visiteur C: enreg-prog-visite C: délivrer-billet > C: OK-deb-visite C: OK-fin-visite

3.3.3) Spécifications fonctionnelles (6)

Formalisme des DTU/DF (*rappel*)

Identificateur DTU/DF	évent. id. du DTU/DF père	Profil Utilisateur	Domaine Fonctionnel
Description des tâches réalisées dans le DTU/DF avec le langage LDTU défini § 4.3.4			lien vers DAU ou DTU/DF

Formalisme des DAU

Identificateur du DAU	identificateur du père (DTU/DF ou DAU)	Rappel : Profil Utilisateur Domaine Fonctionnel
Description des actions réalisée dans le DAU : avec le langage défini § 4.3.4		lien vers DAU ou Commande

3.3.3) Spécifications fonctionnelles (8)

Exemple (*suite*)

□ 3) En déduire la table des utilisateurs par commandes : TU/C

<i>Commandes</i>	Liste des utilisateurs
<i>ou-est-animal?</i>	visiteur, garde, vétérinaire, etc.
<i>nourrir-animal</i>	ingénieur agronome, vétérinaire, garde, etc.
<i>OK-deb-visite</i>	hôtesse, responsable-sécurité,
<i>enregistrer-sos</i>	responsable-sécurité, hôtesse, garde
-----	-----



3.3.3) Spécifications fonctionnelles (9)

□ 4) Description des commandes

- Table de description des commandes (DCOM)
 - description
 - les entrées (*fournis par l'utilisateur*)
 - les retours (*fournis à l'utilisateur par le système cas standard*)
 - la prise en compte des erreurs des utilisateurs
 - les objets concernés
 - qui peut utiliser cette commande

Les retours « normaux » et les retours « en erreur » possèdent des identificateurs (un numéro) qui sera associé à un message dépendant du profil utilisateur et qui sera

Lorsque plusieurs variantes d'une commande sont possibles selon les utilisateurs, deux solutions

- ✓ une seule commande avec identification de l'utilisateur
- ✓ n commandes : une commande par type d'utilisateur

3.3.3) Spécifications fonctionnelles (10)

D-COM : *Exemple*

Commande « *début-sos* »

qui : *responsable sécurité, hôtesse, garde*

- **Description:** permet d'enregistrer un appel au secours
déclenche un certain niveau d'alarme
donne des consignes à la personne qui reçoit l'appel
ouvre dossier pour suivre cet appel

*à vous de
compléter*

Entrées : id-pers-récepteur, id-pers-émetteur, lieu-pers-émetteur,
niveau-urgence, description-situation, -----

Retours : Le système retourne

- accusé de réception (avec un n° de SOS)
- variable-état-retour
- identification du dossier de suivi comportant :
 - *une fiche descriptive du message reçu (date, heure, -----)*
 - *des consignes pour la personne récepteur*
 - *les actions réalisées par le système*

erreurs : (id-pers-récepteur, id-pers-émetteur, lieu-pers-émetteur,
niveau-urgence) incorrect (erreur non fatale)

objet(s) : sos, suivi-sos, personne, lieu, -----

3.3.3) Spécifications fonctionnelles (11)

D-COM : *Exemple*

Commande « *ou-est-animal?* »

Qui : visiteur, garde, vétérinaire, etc.

Description: permet de connaître la dernière localisation

- d'un animal identifié
- ou d'un groupe d'animaux
- La réponse peut être **textuelle** (N° de zone)
ou **graphique** (carte)

à vous de compléter

Entrées : id-demandeur, type-demande⁽¹⁾, id-objet-demandé, type-réponse⁽²⁾

⁽¹⁾ : Individu ou Groupe ⁽²⁾ : Texte ou Carte

Retours : Le système retourne la dernière localisation de cet objet,
la date et l'heure, id-pers-qui-a-localisé, variable-état-retour)

erreurs : paramètres d'entrée incorrects (erreur fatale)
l'objet-demandé n'a pas été localisé (erreur fatale)
le demandeur n'a pas le droit d'utiliser cette commande (erreur fatale)

objet(s) : animal, lieu, personne, -----

Exemple : *Commande : Ou-est-animal?('VTR05',Individu, 'Bouba', 'Texte')*

Retour : E_r=0, R = 'Mont des ours', Z = SE25, P = 56,
date = 1/10/2006, heure = 13h, IPL= 'GRD33',

3.3.3) Spécifications fonctionnelles (12)



□ 5) prise en compte des règles ergonomiques



(cf ch 4)

- Nécessité du guide de style :
 - s'y conformer (noms,)
 - éventuellement le compléter



- Système de commande permettant d'agir sur les objets.
 - **universalité** : existence de commandes à caractère universel
ex. gestion de fichier, édition , etc
 - **cohérence** : commande dans un contexte # sur des objets #
déplacer un paragraphe ou un fichier ou ...
 - **flexibilité** : même but par # méthodes (choix de l'utilisateur)
commandes personnalisées, raccourcis, corrections automatiques
 - **adéquation** : toutes les commandes doivent être utilisées. *Lorsqu'une commande est sous utilisée, en chercher la raison (intérêt, complexité,-visibilité*

3.3.3) Spécifications fonctionnelles (13)



□ 5) *prise en compte des règles ergonomiques (suite)*

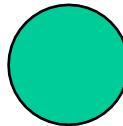
- **Informations associées à une commande** valeurs par défaut :
(analyse préalable des habitudes des utilisateurs)
- **Retours du système**
 - visibles immédiatement pour chaque action
 - conforme à l'attente de l'utilisateur (lié à la métaphore)
 - confirmation de chaque commande dangereuse
- **gestion amicale des erreurs** messages précis
réversibilité
environnement d'aide



3.3.3) Spécifications fonctionnelles (14)

- 6) *prise en compte des règles ergonomiques (fin)*
 - **En conclusion, les choix ergonomiques à ce niveau tendent à rendre le système :**
 - facile à utiliser
 - complet mais pas trop sophistiqué
 - adaptable à l'utilisateur, en tenant compte de son expérience
 - cohérent avec les standards de l'ergonomie informatique
 - ☐ le système de retour doit être facile à interpréter
 - gérer les # types d'erreur
 - permettre leurs corrections
 - **liaison avec l'analyse de l'application (NA)**
 - **Mise à jour :** du glossaire (GLOS)
éventuellement de MSApp

3.3.3) Spécifications fonctionnelles (15)



Synthèse des spécifications fonctionnelles de l'IHM

DS-IHM : Description de la sémantique de l'IHM

PHTU-A Planif. Hiér.de la Tâche Util. Approfondie (*les DAU*) ★

TC/U Tables des commandes / utilisateurs

TU/C Tables des utilisateurs / commande

DCOM Description des commandes ★

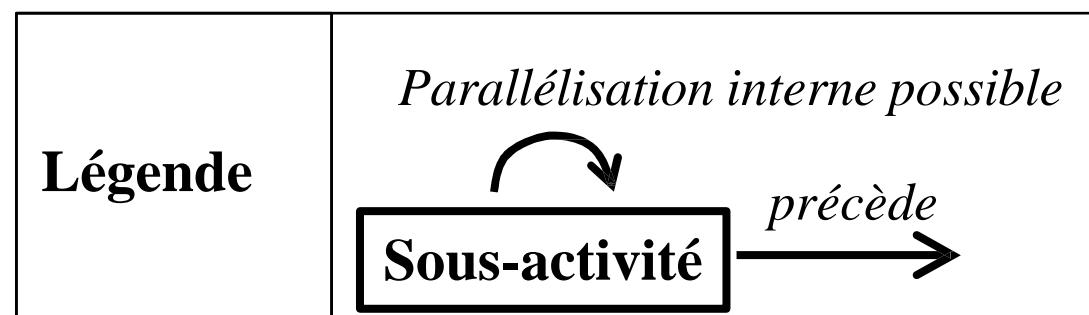
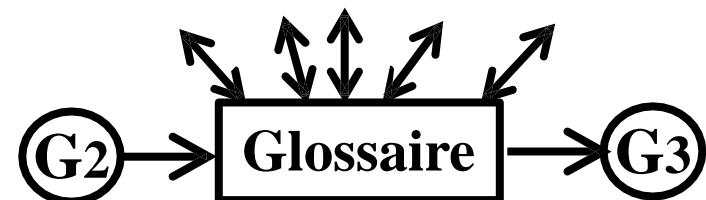
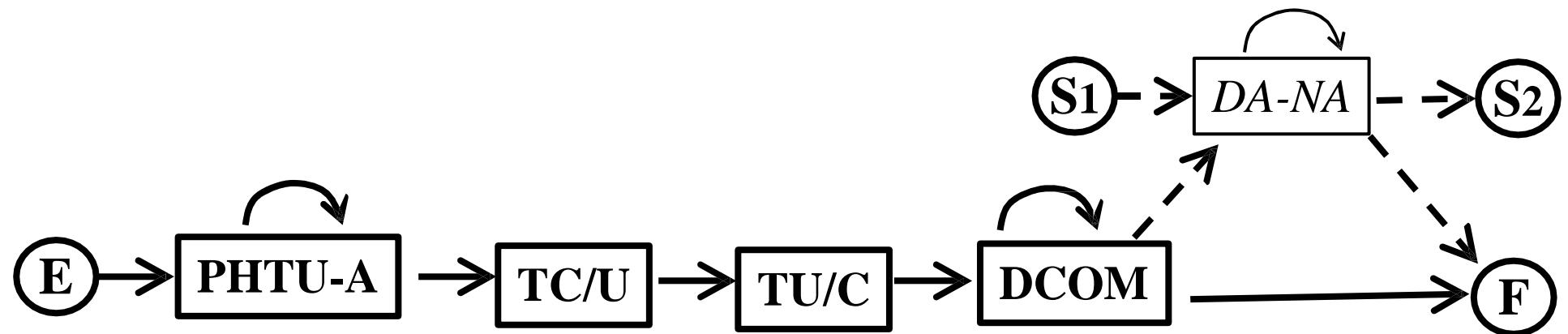
GLOS Mise à jour du glossaire ★

DA-NA Mise à jour du dossier d'analyse (modèle struct) ★

★ : Petites applications

3.3.3) Spécifications fonctionnelles (16)

❑ Ordonnancement des sous-activités des spécifications fonctionnelles

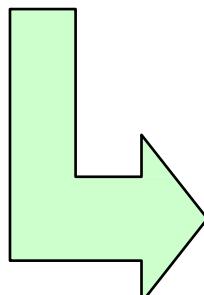


3.3.3) Spécifications fonctionnelles (17)

Spécifications fonctionnelles dans les phases USDP

EP	E	C	T
■	■	■	
■	■	■	
■	■	■	
■	■	■	
■	■	■	
■	■	■	
■	■	■	

- Approfondir la planification hiérarchique de la tâche
- établir la liste des commandes
- décrire les commandes
- prendre en compte les règles ergonomiques
- mettre à jour le modèle structurel du noyau de l'application
- mettre à jour le glossaire



DS-IHM : *Description de la sémantique de l'IHM*

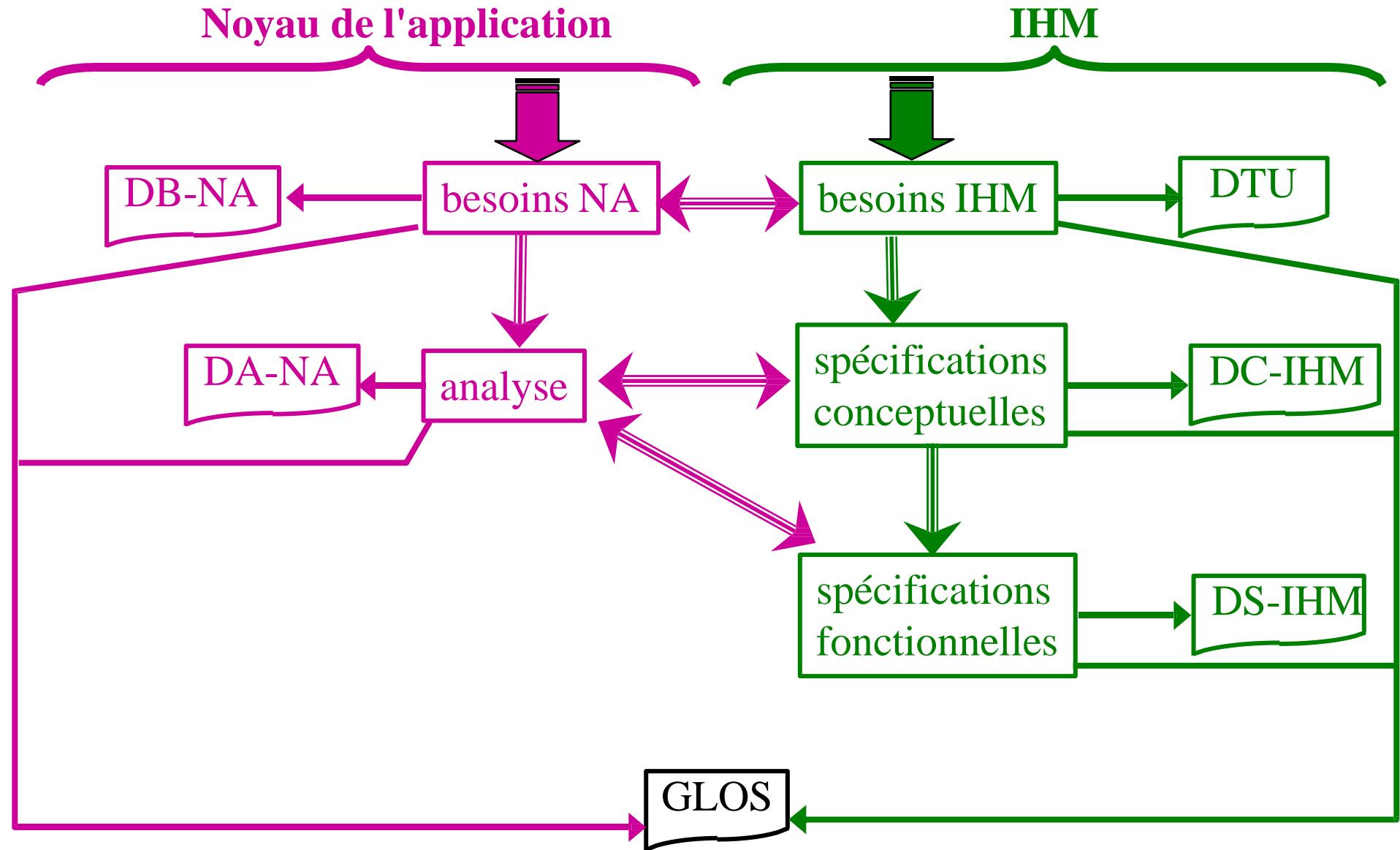
EP: étude préliminaire

E : élaboration

C : Construction

T : transition

3.3.3) Spécifications fonctionnelles (18)



Chapitre 3 : Méthode de conception 00 d'une IHM

- 1) *Présentation de la méthode*
- 2) *Études des besoins pour la CHM*

➤ 3.3) Spécifications de l'IHM

- 1) *introduction*
- 2) *spécifications conceptuelles*
- 3) *spécifications fonctionnelles*

✓ 3.3.4) spécifications syntaxiques

3.3.5) spécifications lexicales

3.4) Réalisation et tests

3.3.4) Spécifications syntaxiques (1)

- **objectif :** définir le dialogue utilisateur-ordinateur
 - qui met en œuvre les actions décrites dans la spécification sémantique.*
 - le style du dialogue
 - la structuration du dialogue
 - la spécification syntaxique des entrées (*commandes*)
 - la spécification syntaxique des sorties (*résultats*)
 - interaction avec les relations entrée et sortie
- **Style du dialogue :** # possibilités définies par rapport :
 - aux contraintes qu'elles imposent aux utilisateurs
 - aux facilités qu'elles offrent en terme : d'expression de but
 - de mémorisation
 - de contrôle par l'utilisateur
 - de guidage par le système
 - d'apprentissage

3.3.4) Spécifications syntaxiques (2)

□ *Style du dialogue (suite)*

1. **dialogue par question réponse**
 - contrôle total au système
 - convient pour les utilisateurs naïfs
 - généralement trop rigide et peu performant
2. **menus** (*permettent d'effectuer des entrées par désignation*)
 - limitent la mémorisation
 - évitent les erreurs lexicales
 - apportent un bon guidage
 - solution idéal pour les utilisateurs naïfs
 - solution acceptable pour les util. expérimentés (si raccourcis)
3. **formulaires** (contrôle alternatif de l'utilisateur & du système)
 - bon compromis pour les entrées de données
 - ex. boîtes de dialogues + boutons*

3.3.4) Spécifications syntaxiques (3)

□ *Style du dialogue (suite)*

4. langage de commande

- contrôle total à l'utilisateur
- très puissant, mais destiné aux experts
- nécessite un apprentissage parfois long

5. touches de fonctions

- sont efficaces en petit nombre
- réservées à des applications spécifiques (conduite de process)
- nécessitent une phase apprentissage

6. langage naturel

- la panacée pour certains ...s'il était au point
- intéressant en sortie pour les messages
- limité en entrée : au niveau syntaxique
 - à une activité particulière
 - lié à la reconnaissance vocale
 - risque de cacophonie

3.3.4) Spécifications syntaxiques (4)

7. style du dialogue (fin)

- **manipulation directe** (*interfaces graphiques*) ou **dialogue objet action** (*interface événementielle*)
 - permet à un utilisateur d'agir directement sur les objets
 - intègre +/- l'ensemble des formes de dialogue précédentes
 - donne le contrôle complet à l'utilisateur
 - n'impose pas l'apprentissage d'un langage
 - solution idéale pour les débutants
 - et pour les experts si l'application s'y prête
 - *c'est le cas de la bureautique*
 - *ce n'est pas toujours le cas en dehors*

3.3.4) Spécifications syntaxiques (5)

□ syntaxe des langages d'interaction (entrée, sortie)

✓ défini par une grammaire

vocabulaire terminal

vocabulaire non terminal

règles de production

symbole d'entrée

✓ langage d'entrée

simple si possible
facile à mémoriser
pas trop verbeux

✓ langage de sortie

présentation des écrans
structuration intra et inter écrans
changement de contexte
menus hiérarchisés
formulaires, etc

doivent tenir compte des règles d'ergonomie

3.3.4) Spécifications syntaxiques (6)

□ Structuration du dialogue

- **échange d'information entre l'utilisateur et le système**
- **structuré en 3 niveaux**
 - ✓ **sous dialogue**
 - *ex. sous dialogue de manipulation du texte*
 - ✓ **la transaction** : élément significatif complet de dialogue
 - partant d'un but élémentaire
 - jusqu'au résultat correspondant
 - *ex . déplacement d'un paragraphe*
 - ✓ **l'interaction** : action élémentaire dans une transaction
 - *désigner le début du §*
 - *désigner la fin du §*
 - *couper*
 - *désigner le nouvel emplacement*
 - *coller*

3.3.4) Spécifications syntaxiques (8)

❖ méthode de spécification

- **différentes méthodes de spécification du dialogue :**

- ✓ graphique :

- des diagrammes d'enchaînement des fenêtres (DEF)
 - des diagrammes de transition d'état des fenêtres (DTEF)
(1 diagramme au plus par fenêtre)

- ✓ textuelle (*adapté aux langages de commande*) :

- spécifications syntaxiques du langage d'entrée
 - spécifications syntaxiques du langage de sortie

❑ ***attention : le dialogue événementiel***

- *est spécifié généralement par une approche graphique*
 - *conduit à une combinatoire difficilement maîtrisable*
 - *doit être réparti sur des objets*

3.3.4) Spécifications syntaxiques (9)

Proposition d'une démarche

➤ choisir le(s) style(s) de dialogue : **StyDia**

- un seul style pour l'ensemble de l'application
- plusieurs styles selon les utilisateurs et/ou les fonctions
- lié à la métaphore

➤ schéma de structuration du dialogue (*grosses applications*) : **Str-dia**

- notation graphiques (cf MAD, § 3.2.4)
- ou notation textuelles (linguistiques)
- dépend du style de dialogue

IHM textuel ou *GAG*

IHM graphique

➤ Spécifications syntaxiques du

- langage d'entrée : **SsyLE**
- langage de sortie : **SsyLS**

GAG

➤ Diagramme d'enchaînement des fenêtres : **DEF**

➤ Diagramme d'état des classes d'objets de l'IHM: **DEC**

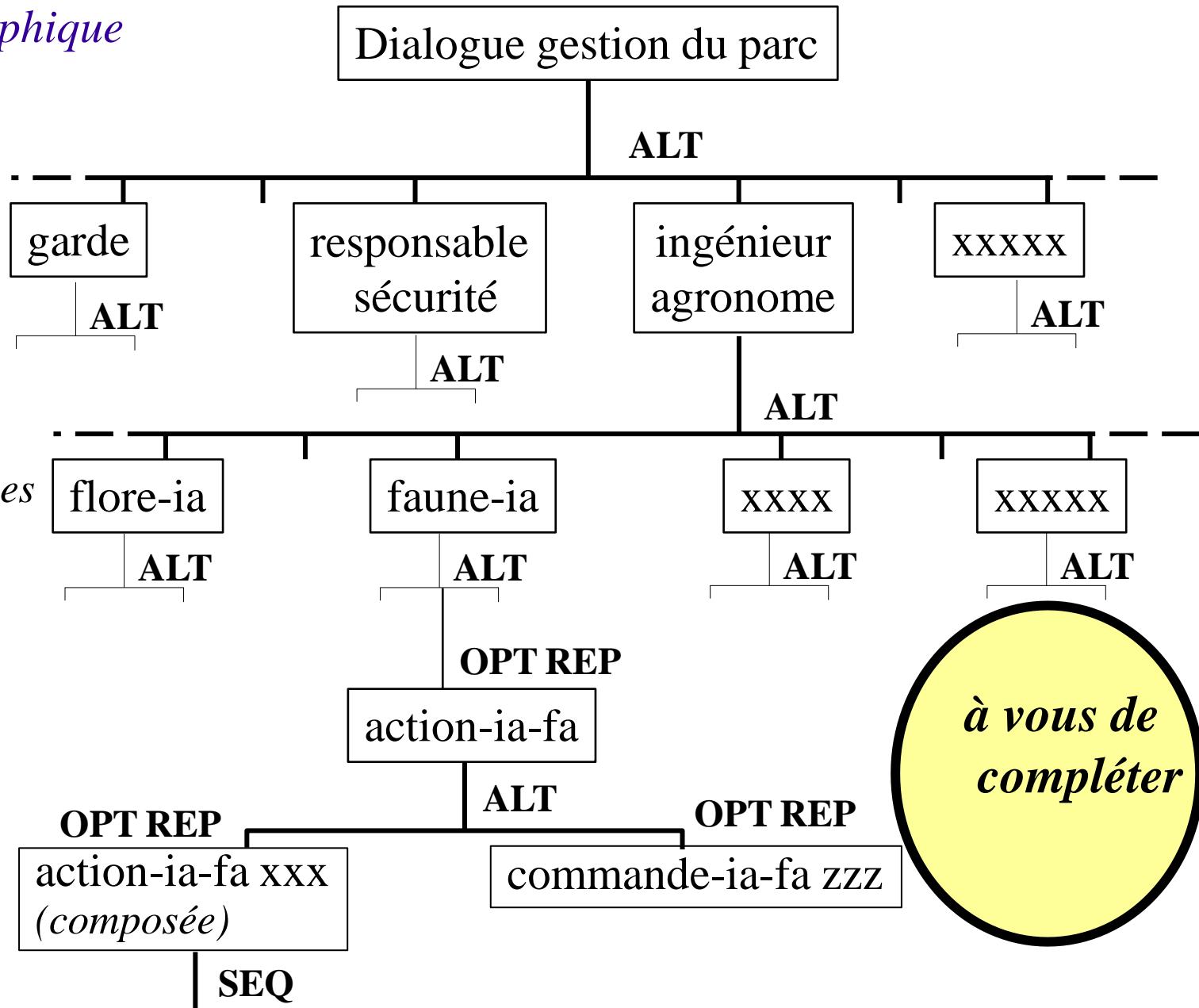
- pour les classes qui ont un comportement complexe
- notations d'Harel

GAG: Pour les grosses applications avec IHM graphiques

*exemple de structuration
de dialogue avec
notations graphique*

3.3.4) Spécifications syntaxiques (10)

*Dialogues
par types
d'utilisateurs*



*Actions et
commandes*

3.3.4) Spécifications syntaxiques (11)

Exemple

□ Style du dialogue

- manipulation directe (*interfaces graphiques*)
- menus et formulaire, un peu d'événementiel

□ Langage d'entrée et langage de sortie (*notations linguistiques*)

Rappel des notations

- Symbole d'entrée : point d'entrée dans le langage (\subset ds VNT)
- Vocabulaire terminal (VT): dans les règles en souligné, défini au niveau lexical, stocké dans le lexique (annexe au dictionnaire)
- Vocabulaire non terminal (VNT) : dans les règles en *italique*, sera stocké dans une table (TVNT - annexe au dictionnaire)
- Règles de production : formalisme ~ BNF

, = séquence	& = agrégat	= choix	{ } = groupe
= 1	+ = 1..n	* = 0..n	? = 0..1

Communication / texte ou / parole (*tél. portable*)

3.3.4) Spécifications syntaxiques (12)

Exemple :
langage d'entrée

- **e-gestion-parc** :: edb-gp, *e-connexion*, *e-dial-tu**, *e-déconnexion*, efn-gp
- *e-connexion* :: edb-cnx, *id-util*, *mot-passe*, ----, efn-cnx
- *e-déconnexion* :: edb-dcnx, *id-cnx*,----, efn-dcnx
- *e-dial-tu* :: *e-dial-ingé-agro* / *e-dial-visiteur* / *e-dial-garde* / ----
- *e-dial-ingé-agro*:: edb-dia, *e-action-df-ia**, efn-
- df-ia :: *e-g-faune-ia* / *e-g-flore-ia* / -----
- -----
- *e-g-faune-ia* :: edb-gfia, *e-quest-faune* / *e-info-faune* /---- efn-gfia,
- *e-quest-faune* :: *e-ou-est-animal* / *e-comment-est-animal* / ---
e-ou-est-animal :: nc-ou-animal, db-par, *id-demandeur*, *type-dem-anim*,
id-objet-demandé, *type-rép-lieu*, ----, fn-par
- *id-demandeur* :: np-demandeur, lien-nom-val, *identificateur*
- *type-dem-anim* :: np-type-dem-a, lien-nom-val, { indiv-animal / gr-animal }
- *type-rép-lieu* :: np-type-rép-lieu, lien-nom-val, { coordonnées / carte | --- }
- -----

à vous de
compléter

X expliquez ces 2 règles

3.3.4) Spécifications syntaxiques (13)

Exemple :
langage de sortie

- **r-gestion-parc** :: rdb-gp, *r-connexion*, *r-dial-tu**, *r-déconnexion*, rfn-gp
- *r-connexion* :: rdb-cnx, *ok*, *id-cnx* / *erreur-cnx*, ----, rfn-cnx
- *r-déconnexion* :: rdb-dcnx, *ok* / *erreur-dcnx*, ----, e-fn-dcnx
- *r-dial-tu* :: *r-dial-ingé-agro* / *r-dial-visiteur* / *r-dial-garde* / ----
- *r-dial-ingé-agro* :: rdb-dia, *r-action-df-ia**, rfn-dia
- *r-action-df-ia* :: *r-g-faune-ia* / *r-g-flore-ia* / -----
- -----
- *r-g-faune-ia* :: rdb-gfia, *r-quest-faune* / *r-info-faune* / --- rfn-gfia,
- *r-quest-faune* :: *r-ou-est-animal* / *r-comment-est-animal* / ---
- *r-ou-est-animal* :: nr-ou-animal, db-par, *id-interaction*,
r-lieu-coordonnées / *r-lieu-carte*, ----, fn-par
- *id-interaction* :: identificateur
- *r-lieu-coordonnées* :: rdb-coo, (*no-rég+*, *no-zone+*, *no-parcelle+*)+, rfn-coo
- *r-lieu-carte* :: rdb-carte, primitives-graphiques+, -----, rfn-carte
- —

à vous de
compléter

X expliquez ces 2 règles

3.3.4) Spécifications syntaxiques (14)

Exemple (suite)

Compléter le glossaire (*cf USDP*)

Symbole	définition	type
gérer-parc	méthode permettant de gérer les attributs de 'parc'	méthode
-----	-----	-----
e-gestion-parc	symbole d'entrée pour le langage d'entrée pour la gestion du parc	VNT
edb-gp	balise début pour e-gestion-parc	VNT
efn-gp	balise fin pour e-gestion-parc	VNT
e-connexion	commande de connexion d'un utilisateur (entrée)	VNT
-----	-----	-----
e-g-faune-ia	actions du domaine fonctionnel 'gestion de la faune' destinées aux ingénieurs agronomes (entrée)	VNT
-----	-----	-----
nc-ou-animal	nom de la commande permettant de localiser un animal ou un groupe d'animaux (entrée)	VNT
db-par	balise début de liste de paramètres dans une commande	VNT
fn-par	balise fin de liste de paramètres dans une commande	VNT
-----	-----	-----

à vous de
compléter

3.3.4) Spécifications syntaxiques (15)

Exemple (*suite*)

□ Diagramme d'enchaînement des fenêtres (DEF)

- L'application est structurée en hypergraphe.
- Chaque noeud de l'hypergraphe est associé à :
 - une icône inspirée par l'activité correspondante du parc
 - une bulle
 - un texte court décrivant la fonction de ce noeud (quelques lignes)
 - une fenêtre
- **Noeud élémentaire**
 - correspond à une transaction
 - est associé à une fenêtre contenant un ou plusieurs objets graphiques permettant à l'utilisateur de construire sa commande et d'en visualiser le résultat
- **Noeud composé :**
 - correspond à une famille de transactions
 - est associé à une fenêtre contenant une portion de l'hypergraphe
un menu.

GUI

à compléter

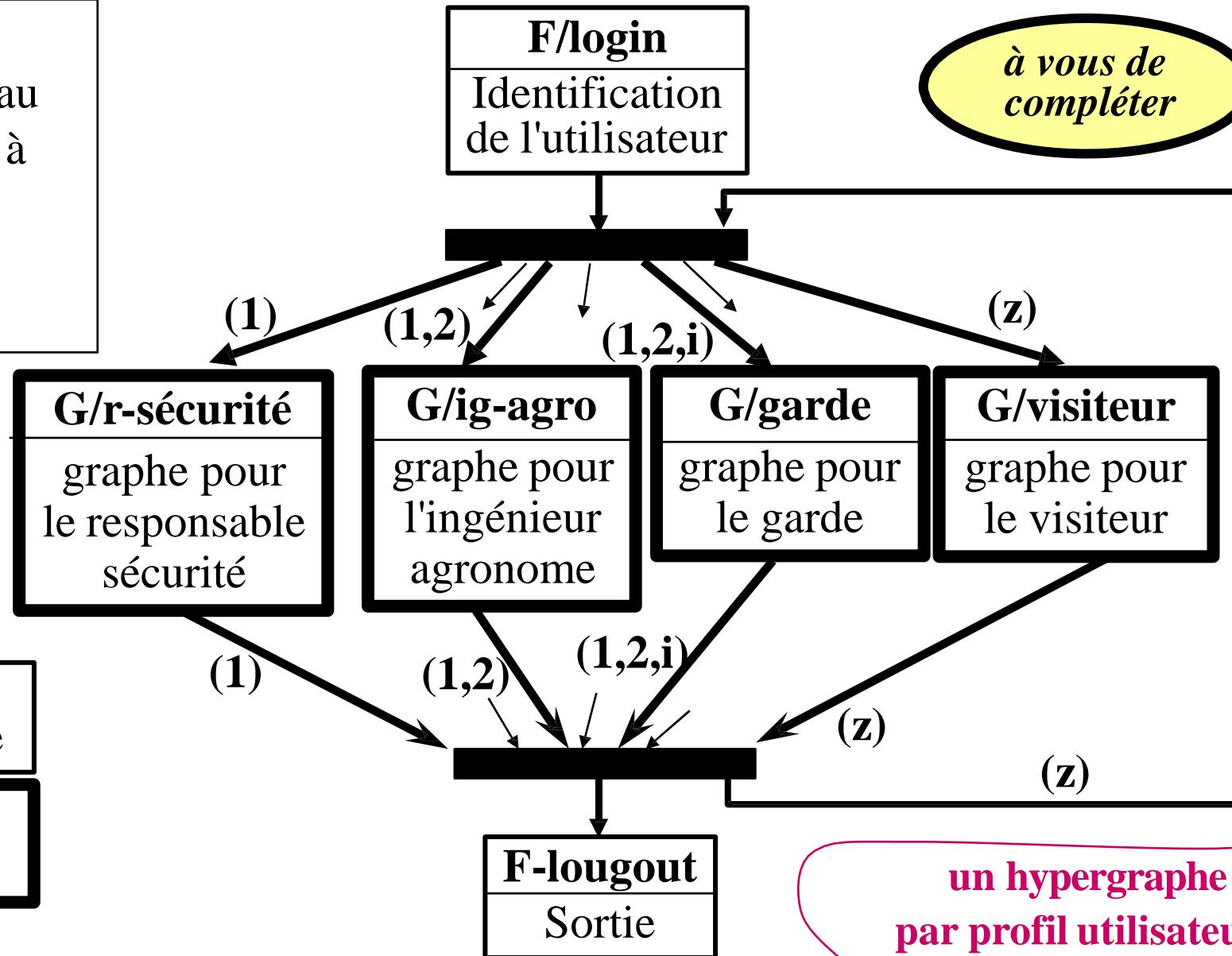
3.3.4) Spécifications syntaxiques (16)

Exemple (suite)

□ Diagramme d'enchaînement des fenêtres (DEF)

Arc autorisé 1: au resp. séc. 2: à l'IG-agro
 - : -----
 i : au garde z : à tous

Nœud élémentaire
Nœud composé

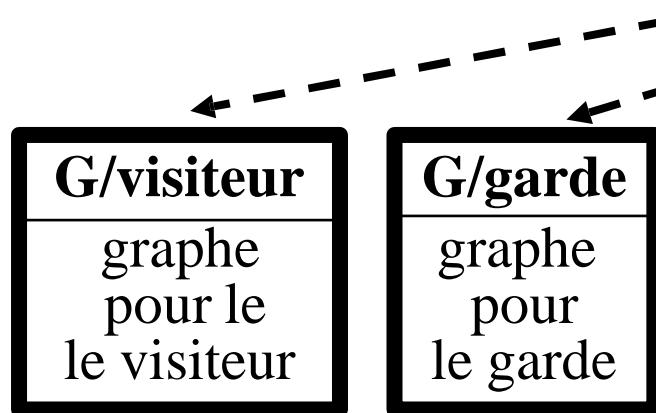


3.3.4) Spécifications syntaxiques (17)

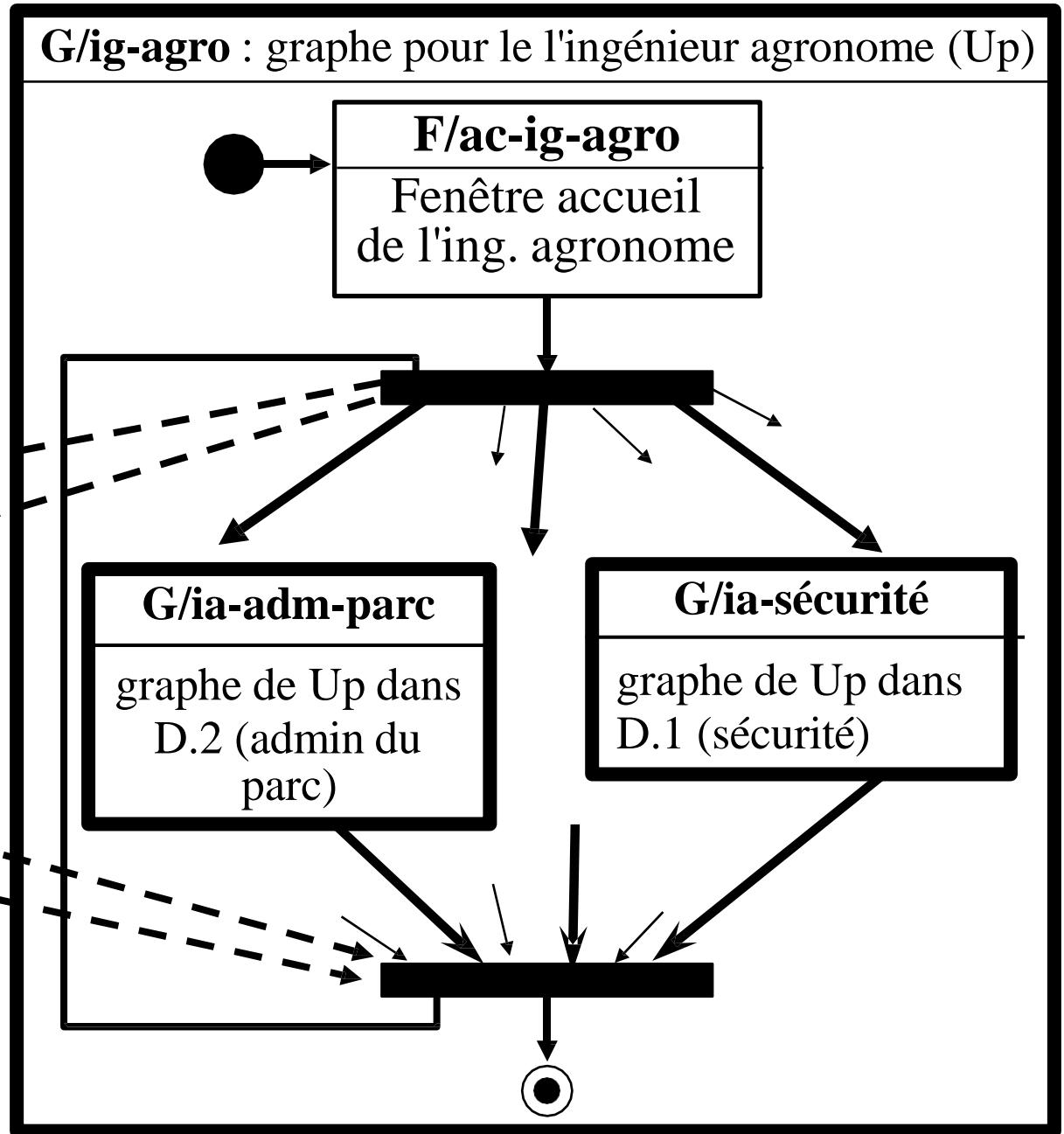
Exemple (suite)

DEF

à vous de compléter



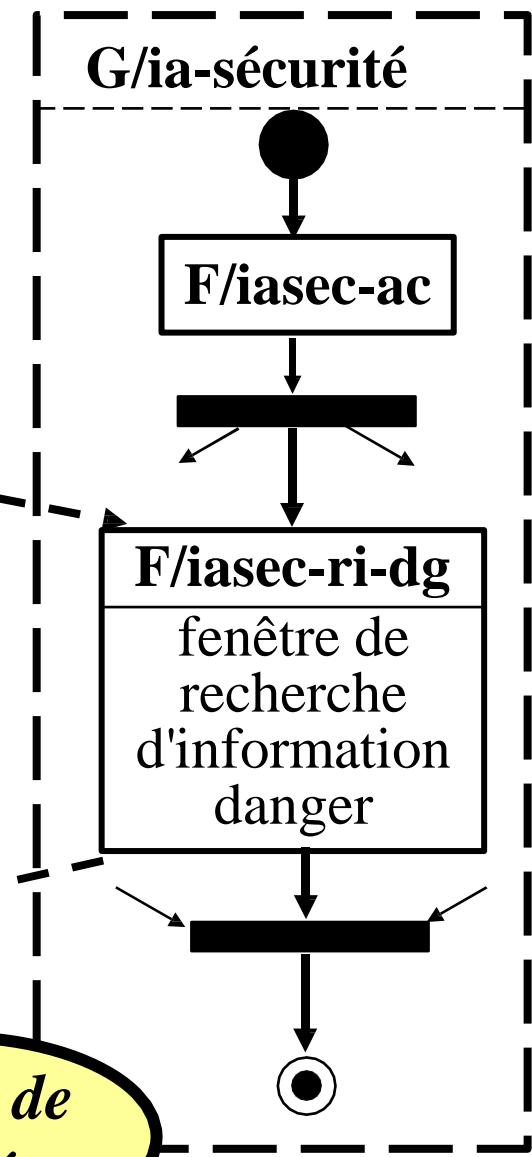
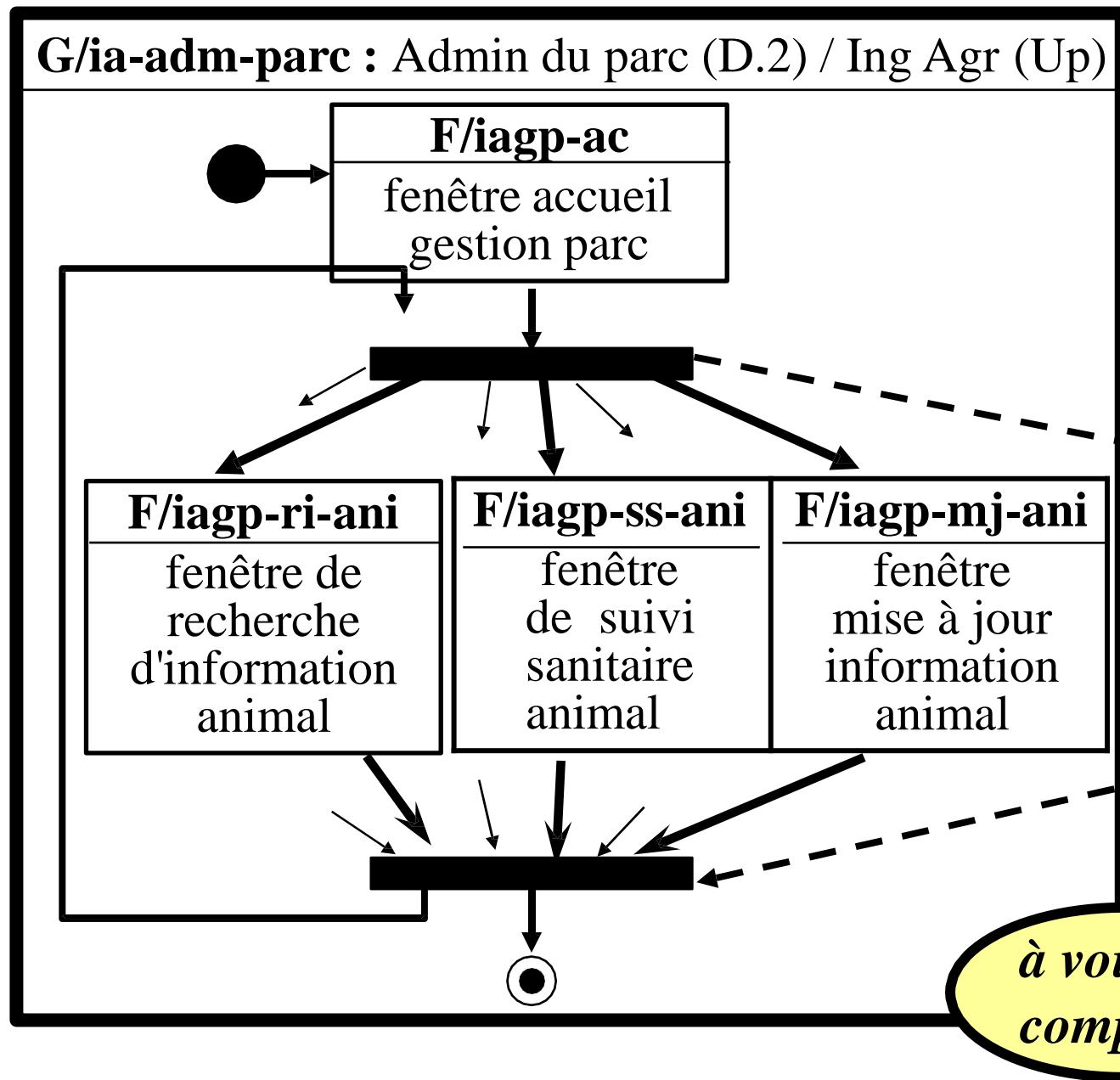
un graphe
par utilisateur / DF



3.3.4) Spécifications syntaxiques (18)

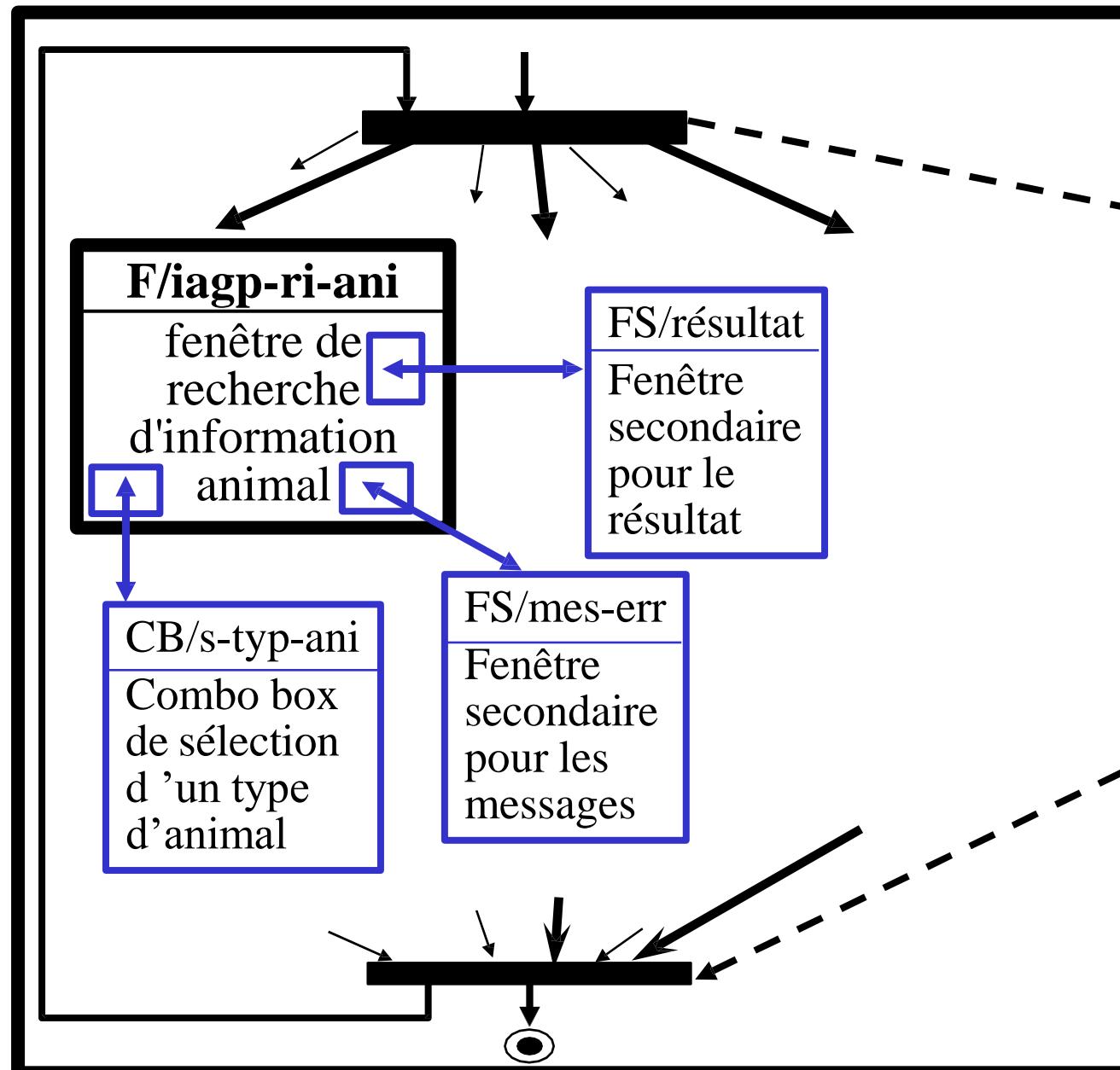
Exemple (*suite*)

DEF



3.3.4) Spécifications syntaxiques (19)

Exemple (suite)
DEF



*à vous de
compléter*

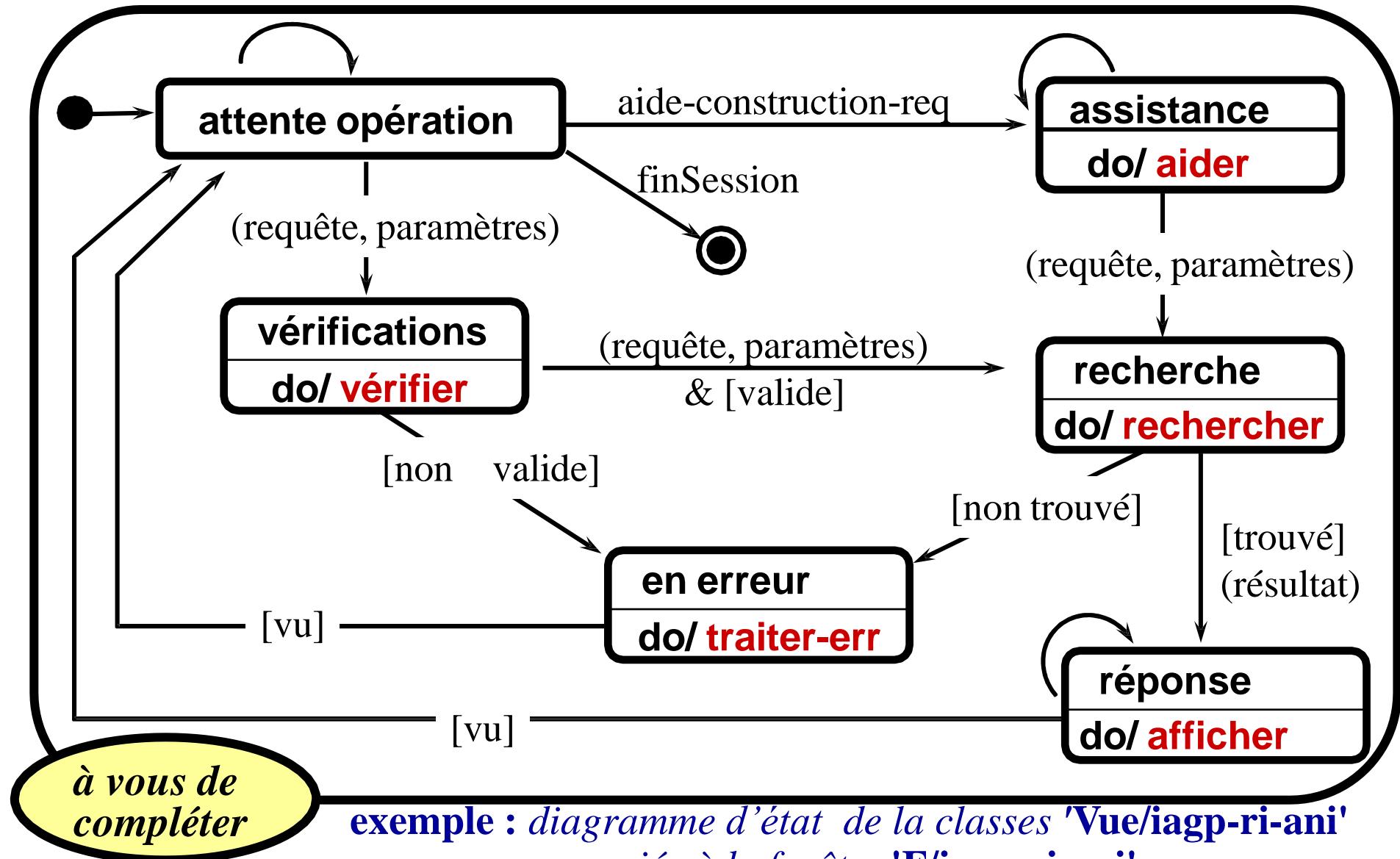
*Les fenêtres
secondaires*

Une
fenêtre
principale
pour une tâche
de haut niveau

3.3.4) Spécifications syntaxiques (20)

Exemple de DEC

Diagramme d'état des classes de l'IHM (*Maj du dos. de concept. NA*)



Synthèse des spécifications syntaxique de l'IHM

DSy-IHM : Description syntaxique de l'IHM

StyDia	style du dialogue
StrDia	schéma de structuration du dialogue
SSyLE	spécification syntaxique du langage d'entrée
SSyLS	spécification syntaxique du langage de sortie
DEF	diagramme d'enchaînement des fenêtres
DEC	diagrammes d'état des princ. classes de IHM

GLOS	Mise à jour du glossaire
DC-NA	Mise à jour du dossier de conception du IHM

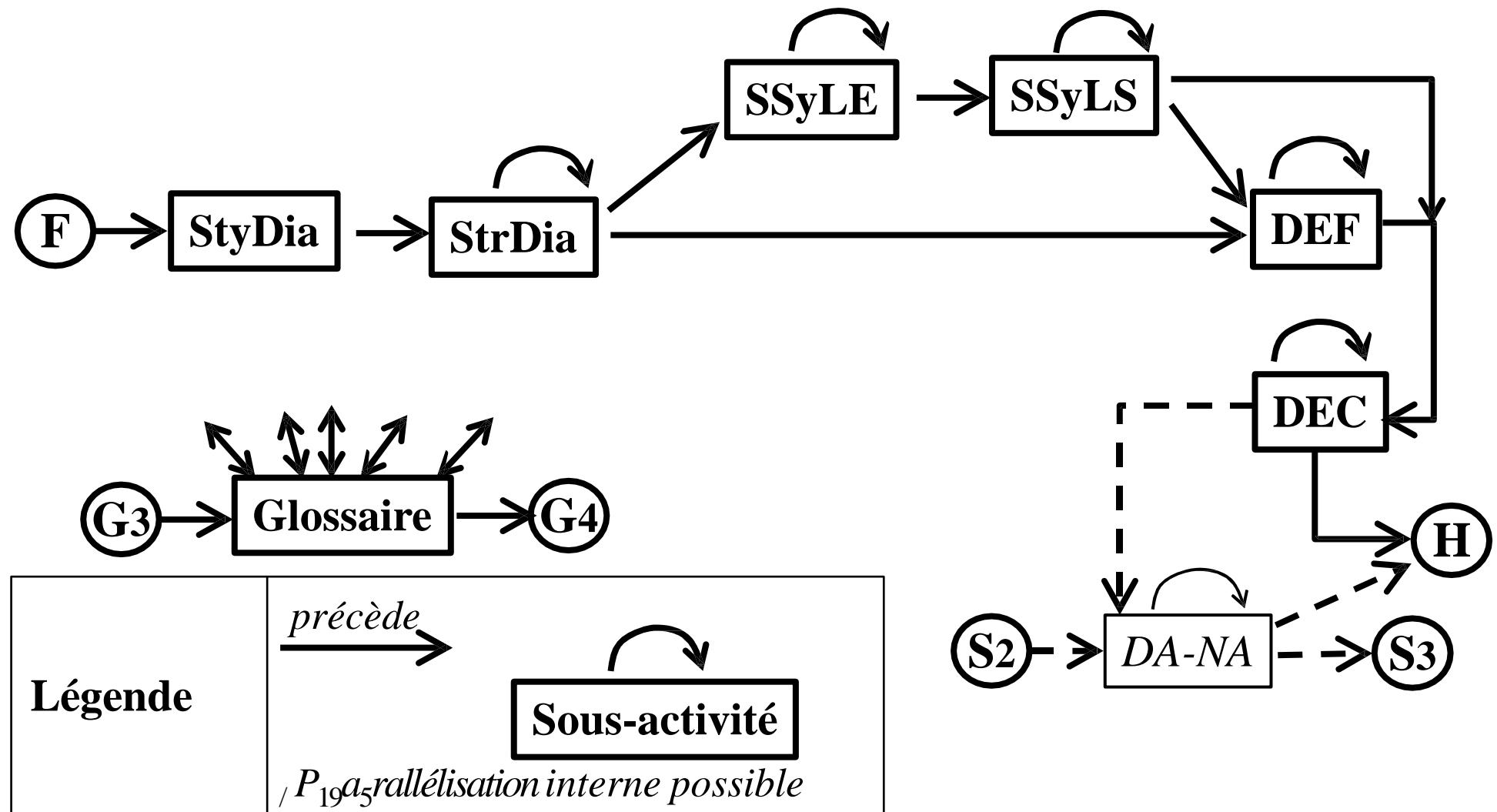
★ : Petites applications
(IHM graphique)

★ : Petites applications
(IHM textuel ou parlé)

(si interface graphique)
(si interface textuel ou parlé)

3.3.4) Spécifications syntaxiques (22)

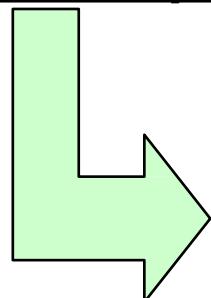
❑ Ordonnancement des sous-activités des spécifications syntaxiques



3.3.4) Spécifications syntaxiques (23)

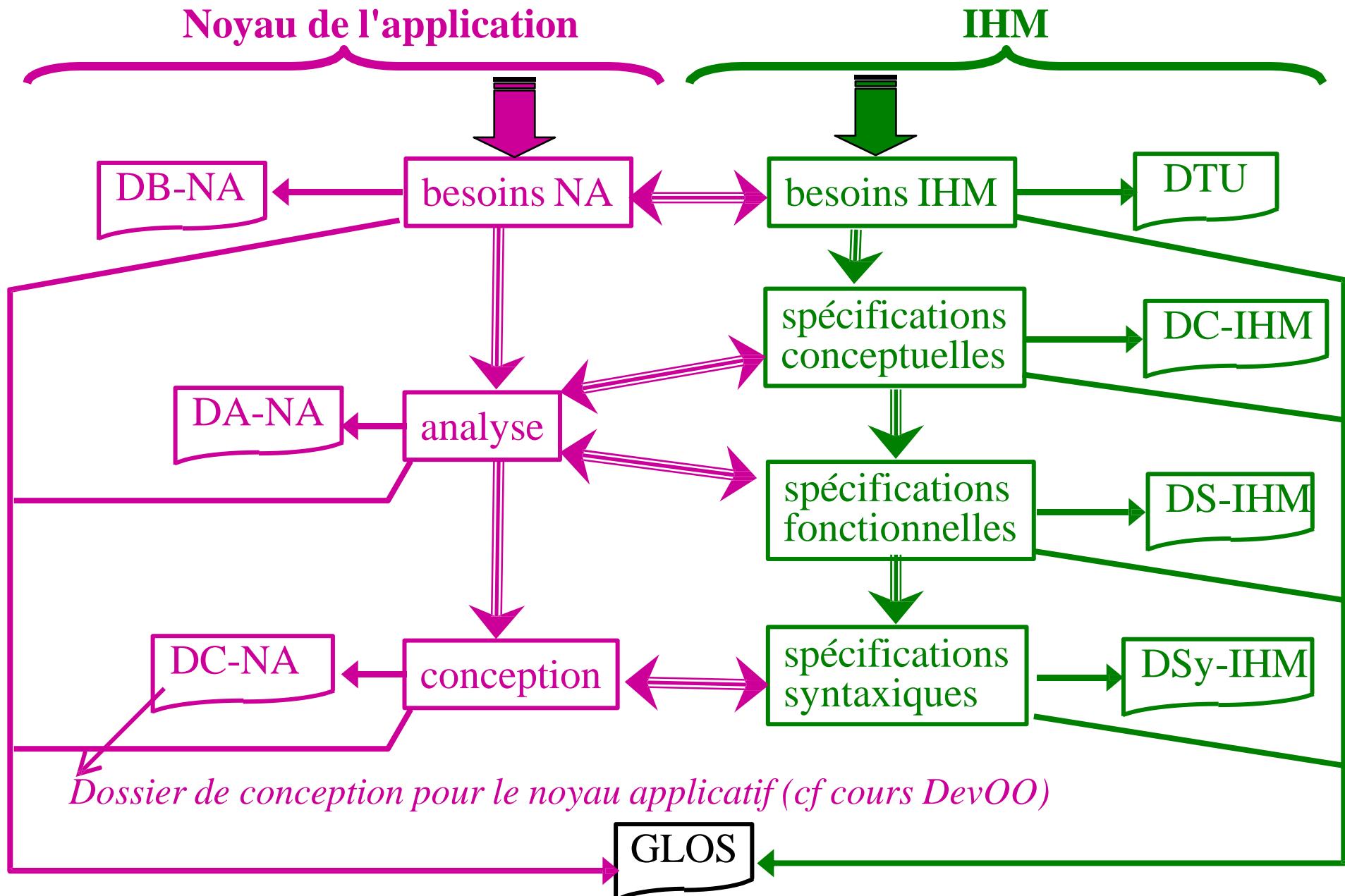
Spécifications syntaxiques dans les phases USDP

EP	E	C	T	
■	■			□ Définir le style du dialogue
■	■	■		□ Définir la syntaxe des langages d'entrée
■	■	■		□ Définir la syntaxe des langages de sortie
■	■	■		□ Définir la structure du dialogue
■	■	■		□ Réaliser le diagramme d'enchaînement des fenêtres
■	■	■		□ Réaliser le diagrammes d'état des classes de l'IHM
■	■	■		□ Mettre à jour le glossaire



DSy-IHM *Description syntaxique de l'IHM*

3.3.4) Spécifications syntaxiques (25)



Chapitre 3 : Méthode de conception 00 d'une IHM

- 1) *Présentation de la méthode*
- 2) *Études des besoins pour la CHM*

➤ 3.3) Spécifications de l'IHM

- 1) *introduction*
- 2) *spécifications conceptuelles*
- 3) *spécifications fonctionnelles*
- 4) *spécifications syntaxiques*

✓ 3.3.5) spécifications lexicales

- 3.4) Réalisation et tests

3.3.5) Spécifications lexicales (1)

❖ Elle détermine ce qui est immédiatement visible de l'utilisateur

:

- **les dénominations**
 - dénomination des commandes, des items, des menus,
 - formulation des messages, de l'aide
- **la représentation graphique**
 - dessin des fenêtres
 - choix des dessins et des icônes
 - choix des mises en valeur (*surbrillance, clignotement, ...*)
 - typographie (*police, corps, styles*)
- **la pragmatique d'utilisation**
- **le codage :**
 - les couleur
 - les boutons
 - les abréviations

lexique

Intention
Contrôle
Action
Réponse

En conformité
avec le guide
de style

3.3.5) Spécifications lexicales (2)

❖ Il faut appliquer les règles d'ergonomie :

- Utiliser la couleur pour :
 - montrer la structure logique
 - mettre en relation des éléments dispersés sur l'écran
 - attirer l'attention
 - différencier les statuts de l'informations
- Attention ✓ éviter les couleurs systématiques
 - ✓ limiter le nombres de couleurs (n < 8)
 - ✓ le noir et blanc est très souvent optimum
 - ✓ choisir des couleurs significatives
- tenir compte
 - ✓ des archétypes en vigueur rouge = danger
 - ✓ orange = attention
 - ✓ vert = OK : continuez
 - ✓ déficiences physiologiques : 6 à 8 % de daltoniens
 - ✓ technologies : *impression N&B de documents en couleurs*

3.3.5) Spécifications lexicales (3)

Exemple (*suite*)

dénomination : lexique du vocabulaire terminal (LxqVT)

concept vnt	symbole vt	commentaire
edb-gp	[?GParc]	balise début du dialogue de gestion du parc
efn-gp	[?GParc/]	balise fin du dialogue de gestion du parc
edb-dia	[?D-IAgro]	balise début du dialogue ingénieur agronome (IA)
efn-dia	[?D-IAgro/]	balise fin du dialogue IA
edb-gfia	[?DIA-faune]	balise déb. du dial. IA pour la gestion de la faune
efn-gfia	[?DIA-faune/]	balise fin du dial. IA pour la gestion de la faune
nc-ou-animal	?ou-est-animal	nom de la com. permettant de localiser un animal
db-par	(balise début de liste de paramètres ds commande
fn-par)	balise fin de liste de paramètres ds commande
np-demandeur	demandeur	nom du paramètre identifiant le demandeur
lien-nom-val	=	lien entre le nom d'un paramètre et sa valeur
np-type-dem-a	type-demande	nom du param. identifiant le type de demande (anim.)
np-type-dem-a	type-rép-lieu	nom du param. identifiant le type de réponse (anim.)
-----	-----	-----

3.3.5) Spécifications lexicales (4)

exemple
langage d'entrée

[?GParc]

[?D-IAgro]

[?DIA-faune]

?Ou-est-animal (demandeur = VTR05',
type-demande = Individu,
nom-animal = Bouba,
type-rép-lieu = coordonnées)

[?DIA-faune/]

[?D-IAgro/]

[?GParc/]

à vous de
compléter

X : expliquez cette commande

3.3.5) Spécifications lexicales (5)

exemple
langage de sortie

[!GParc]

[!D-IAgro]

[!DIA-faune]

!Ou-est-animal (id-requête = 0001458365

 demandeur = VTR05,

 nom-animal = Bouba,

 E_R = 0,

 <----- E_R = 0 : Retour OK
 coordonnées = rég: SE, zone : 54, parc. : 133

 date = 01/10/2008)

[!DIA-faune/]

[!D-IAgro/]

[!GParc/]

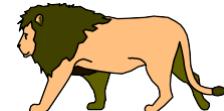
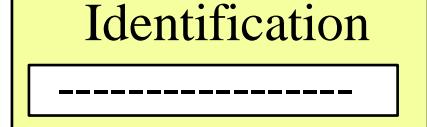
à vous de
compléter

X : expliquez cette réponse

3.3.5) Spécifications lexicales (6)

exemple :

❖ pour les IG : lexique des objets graphiques (LxqOG)

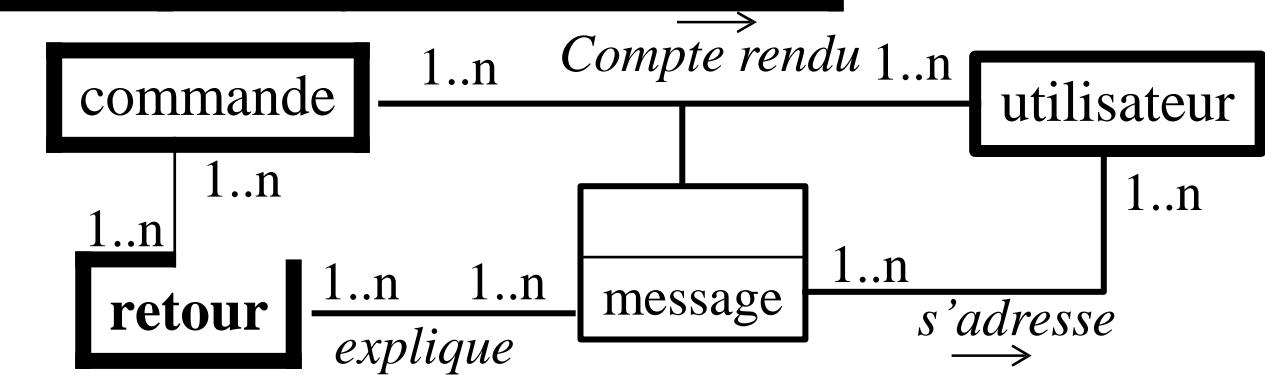
identificateur	description	dessin
icô-parc	icône du parc	
icô-faune	icône représentant la faune	
icô-fri	icône de la fonction recherche d'information	
bdial-ident	boîte de dialogue pour identifier un objet du parc (<i>animal, végétal, ----</i>)	
-----	-----	-----

3.3.5) Spécifications lexicales (7)

□ Définition des tables de messages / type d'utilisateur (TMesU)

N°	Texte/ util-A	N°	Texte/ util-B
0	<i>OK</i>	100	<i>that's right</i>
1	<i>utilisateur non identifié</i>	101	<i>who are you ?</i>
2	<i>demandeur non autorisé</i>	102	<i>you can't</i>
...		
30	<i>identification incorrecte</i>	130	<i>bad identification</i>
31	<i>impossible de localiser</i>	131	<i>I can't find</i>
32	<i>localisation incorrecte</i>	132	
...			
i	<i>date incorrecte</i>	j	
...			

exemple :
table des messages pour des utilisateur de type A et B



à vous de compléter

3.3.5) Spécifications lexicales (8)

exemple

□ Définition de la table TCUM (Commande / Utilisateur / Messages)

Commande	E_R	profil utilisateur		
		A	B	S
Ou-est-animal?	1	0	100	...
	2	1	101	
	3	2	102	
	4	30	130	
	5	31	131	

Animal-est-à	1	0	100	
	2	1	101	
	3	2	102	
	4	30	130	
	5	32	132	

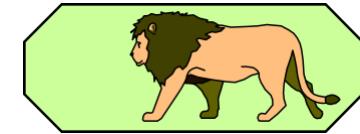
*à vous de
compléter*

3.3.5) Spécifications lexicales (9)

exemple :

□ Fiche descriptive de fenêtre (FDFen)

'Recherche d'information sur la faune'

- ❖ identificateur de la fenêtre : **F/iafa-ri-ani**
- ❖ icône associé : **icô-ri-faune** ----- → 
- ❖ titre : **tit-fen** ----- → *'Recherche d'information sur la faune'*
- ❖ contexte : **ctx-fen** ----- → *G/ig-agro> G/ia-faune ; -----*
- ❖ états : **état-fen**
 - 1 : *non active*
 - 2 : *requête en cours de construction sans assistance*
 - 3 : *en attente du résultat*
 - 4 : *affichage du résultat correct*
 - 5 : *affichage d'un message d'erreur*
 - : -----

*Cf le diagramme d'état de la classe
d'objet de l'IHM correspondant*

*à vous de
compléter*

3.3.5) Spécifications lexicales (10)

exemple (suite)

□ *Fiche descriptive de fenêtre (suite)*

■ **composition :**

- choix de l'objet
 - animal individuel ou groupe d'animaux : **br-chx-ind**, **br-chx-grp**
 - identificateur de l'objet : **bdial-ident** (*si demande d'assistance : liste des (groupes) animaux combo-box*)
- sous-fonction choisie : **bdial-fonct**
 - tout objet faune : localisation, présentation, alimentation, ----
 - individu : pedigree, photographie, ----
 - si demande d'assistance : liste des fonctions (*combo-box*)
- choix de la présentation du résultat (*case à cocher*)
 - texte : **cch-rés-txt**
 - graphique : **cch-rés-gra**
 - photo : **cch-rés-pho**
 - vidéo : **cch-rés-vid**

à vous de compléter

3.3.5) Spécifications lexicales (11)

exemple (suite)

- *Fiche descriptive de fenêtre (suite)*

- *composition (suite):*

- ✓ zone pour les fenêtres secondaire
 - ✓ aide à la formulation de requête: **cbx-fonct**, **cbx-obj1**, ... (*checkbox*)
 - ✓ ou présentation du résultat : **fs-rés**
 - ✓ ou édition de messages : **fs-mes**
- ✓ barre de commande : boutons
 - ✓ début, précédent, suite : **btn-déb**, **btn-préc**, **btn-suite**, ...
 - ✓ fermer, annuler, aide, ...
 - ✓ OK : **btn-OK**
- ✓ barre d'état
 - ✓ nom du demandeur et type d'utilisateur
 - date et heure
 - identification de la fenêtre et état
 - coût, etc.

*à vous de
compléter*

3.3.5) Spécifications lexicales (12)

□ D T D simplifiée d'une fiche descriptive d'une fenêtre

```
<DOCTYPE .....
-----
<ELEMENT fenetre_princ (identification, composition, barre_com, barre_etat)>
  <ATTLIST fenetre_princ id_fp ID REQUIRED,
    etat_fen CDATA IMPLIED>
<ELEMENT identification (titre, icone, contexte)>
<ELEMENT (titre, contexte) (# PCDATA)>
<ELEMENT icone EMPTY>
  <ATTLIST icone image ENTITY REQUIRED>
<ELEMENT composition (obj_util+)>
<ELEMENT obj_util (obj_util_comp | obj_util_elem)>
  <ATTLIST obj_util id_ou ID REQUIRED,
<ELEMENT obj_util_comp (fenetre_secondaire* | composition*)>
<ELEMENT obj_util_elem (bouton | bouton_radio | case_a_cocher | combo-box |
  boite_dialogue | icone | ...)>
-----
<ELEMENT barre_com (bouton+)>
<ELEMENT barre_etat ((icône| champ_texte)+)>
-----
]>
```

3.3.5) Spécifications lexicales (13)

■ Instance simplifiée

```
<DOCTYPE ....
-----
<fenetre_princ if_fp = "F/iafa-ri-ani" etat_fen = "0" >
  <identification>
    <titre> Recherche d'information sur la faune </titre>
    <icône image= "----"/>
    <contexte> G/ig-agro/ia-faune/--- </contexte> </identification>
  <composition>
    <obj_util id_ou = "----" >
      -----
      </composition>
      -----
      <barre_com>
        <bouton> -----</bouton>
        <bouton> -----</bouton>
      -----
      </barre_com>
      <barre_etat>
        <champ_texte> .... </champ_texte>
        -----
        <champ_texte> .... </champ_texte>
      </barre_etat>
      -----
    </fenetre_princ >
```

3.3.5) Spécifications lexicales (14)

□ Quelques remarques

- Le modèle de fenêtre peut être décrit avec un schéma XML
 - plus précis
 - instance identique
- L'instance correspond à une description logique de la fenêtre
 - une feuille de style + traitements (XSL ou ...) sont donc nécessaire pour la finaliser
 - ce qui permet de s'adapter à tout type de terminal
- Le mécanisme (description logique + traitements) peut être généralisé à niveau de la totalité de l'IHM
- Il existe différents modèles réutilisables définis par groupes de travail certains sont identifiés par le W3C et accessibles via internet
 - Certains peuvent être considérés comme des « quasi standard » (cf chap 6)
 - □ XUL, UIML, XAML, ...

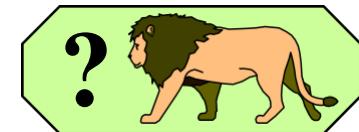
3.3.5) Spécifications lexicales (15)

exemple : dessin de fenêtre
état : 0 (non actif)

Parc Européen



Recherche d'information sur la faune



individu
groupe

Identification

nom

information recherchée :

fonction

résultats :

texte

graphique

photographie

vidéo

début

précédent

suite

fermer

annuler

aide

--

--

OK

nom-demandeur

type-u

Date / heure

id-fenêtre

état

coût

(-----)

*zone pour les fenêtres secondaires
pour l'aide à la formulation de
requête ou à la présentation du
résultat*

*ou à l'édition de
messages ou à l'édition
de l'aide*

(message)

3.3.5) Spécifications lexicales (16)

exemple : dessin de fenêtre
état : 2 (assistance)

Recherche d'information sur la faune

?

The screenshot shows a user interface for searching animal information. At the top left is a landscape photograph of a lake and mountains. To its right is a green octagonal icon containing a lion's head with a question mark above it. Below the photograph are two radio buttons: 'individu' (selected) and 'groupe'. A large yellow box labeled 'Identification' contains a text input field with 'nom'. To the right of these are three nested menus. The innermost menu lists 'animaux du parc', 'mammifères', 'oiseaux', and 'reptile'. The middle menu lists 'mammifères', 'chamois', 'lions', and 'ours'. The outermost menu lists 'ours', 'Bouba', 'Frisquette', and 'Nonor'. A large yellow button at the bottom center says 'choisissez dans la liste, cliquez avec la souris'. At the bottom are buttons for 'début', ' précédent', 'suite', 'fermer', 'annuler', 'aide', '...', 'OK', and status bars for 'JM Pinon', 'Ing. Agro', '01-04-2001 :12-48', 'F/iafa-ri-ani', '2', '5', and '(- - -)'.

individu groupe

Identification nom

information recherchée : fonction

résultats : texte graphique photographie vidéo

choisissez dans la liste, cliquez avec la souris

début précédent suite fermer annuler aide OK

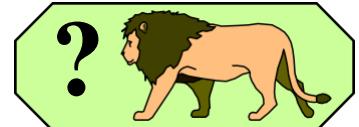
JM Pinon Ing. Agro 01-04-2001 :12-48 F/iafa-ri-ani 2 5 (- - -)

3.3.5) Spécifications lexicales (17)

exemple : dessin de fenêtre
état : 2 (assistance)

Recherche d'information sur la faune

?



individu groupe

Identification

Frisquette

information recherchée : fonction

fonctions régime présentation localisation

résultats : texte graphique photographie vidéo

choisissez une fonction dans la liste, cliquez avec la souris

début précédent suite fermer annuler aide OK

JM Pinon Ing. Agro 01-04-2001 :12-49 F/iafa-ri-ani 2 6 (----)

3.3.5) Spécifications lexicales (18)

exemple : dessin de fenêtre
état : 4 (résultat)

Recherche d'information sur la faune

?

individu

groupe

Identification

Frisquette

information recherchée :

présentation

résultats :

texte

graphique

photographie

vidéo

Frisquette est une ourse de la famille des grizzli. Elle est âgée de 4 ans et n'a pas encore eu d'oursons. -----

pour connaître où se trouve Frisquette actuellement choisissez ' information recherchée ' = ' localisation '

début précédent suite fermer annuler aide --- --- OK

JM Pinon Ing. Agro 01-04-2001 :12-52 F/iafa-ri-ani 4 10 (----)

3.3.5) Spécifications lexicales (19)

❖ Modèle ICAR de la fenêtre : *Recherche d'information sur la faune*

Intention	Contrôle	Action	Réponse
1) choisir individu ou groupe	1) b-radio 'br-chx-ind' ou 'br-chx-grp'	clic	■ noircit le bouton correspondant
2) Identifier objet sans assistance	1) b-dialogue 'bdial-ident' 2) bouton 'OK'	saisie clic	■ si OK : 'OK' dans fsmes sinon : message d'erreur et assistance (cf 3)
3) Identifier objet avec assistance	1) b-dialogue 'bdial-ident' 2) bouton 'OK'	saisie " clic	■ choix guidé par combo-box en cascades (cbx-obj-1, ---)
4) choisir la fonction sans assistance	1) b-dialogue 'bdial-fonct' 2) bouton 'OK'	saisie clic	■ si OK : 'OK' dans fsmes sinon : message d'erreur et assistance (cf 5)
5) choisir la fonction avec assistance	1) b-dialogue 'bdial-fonct' 2) bouton 'OK'	saisie clic	■ choix guidé par combo-box cbx_fonct

3.3.5) Spécifications lexicales (20)

□ Modèle ICAR de la fenêtre : *Recherche d'information sur la faune* (suite)

Intention	Contrôle	Action	Réponse
6) choisir le type du résultat	1) cch-rés-txt 2) cch-rés-gra 3) cch-rés-pho 4) cch-rés-vid 5) bouton 'OK'	✓ clic ✓ clic ✓ clic ✓ clic ✓ clic	coche la case correspondante <u>si</u> OK : 1) lance la visualisation du résultat dans fs-rés 2) messages pour guider ds fs-mes <u>sinon</u> : message d'erreur dans fs-mes

à vous de compléter

raccourci : *RC* = clic sur le bouton *OK*

3.3.5) Spécifications lexicales (21)

Synthèse

Synthèse des spécifications lexicales de l'IHM

DLx-IHM : Description lexicale de l'IHM



LxqVT

lexique du vocabulaire terminal (*langages E / S*)



TMesU

tables des messages / type d 'utilisateur

TCUM

tables Commande / Utilisateur / Message

(*si interface graphique*)

LxqOG

lexique des objets graphiques

FDFen

fiches descriptives des fenêtres principales (1 / fen.)

DesFen

dessins des fenêtres principales (n / fenêtre princ.)

TICAR

tableau ICAR des fenêtres principales (1 / fen.)



GLOS

Mise à jour du glossaire



DC-NA

Mise à jour du dossier de conception du **A**



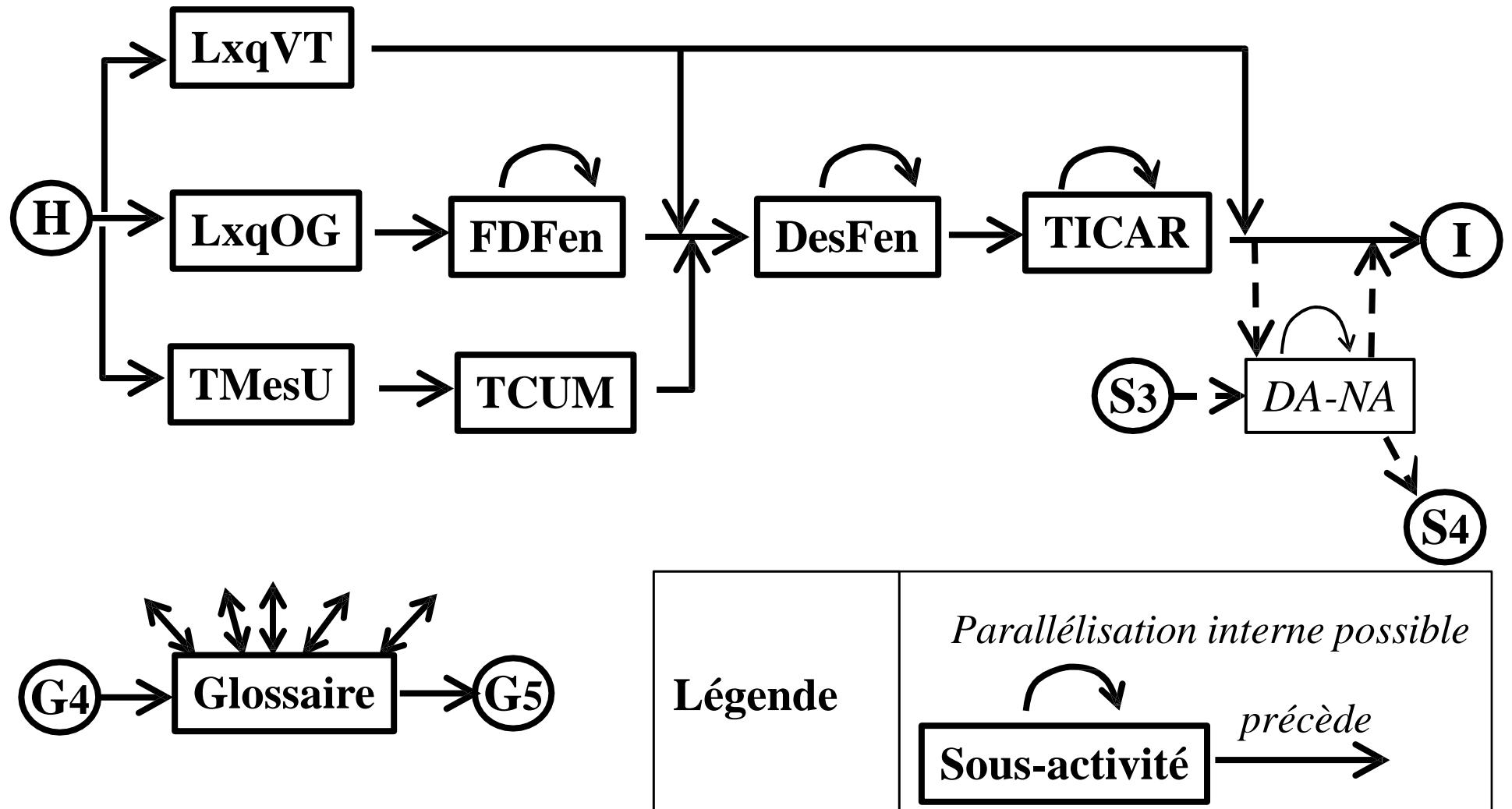
: Petites applications
(*IHM textuel ou parlé*)



: Petites applications
(*IHM graphique*)

3.3.5) Spécifications lexicales (22)

❑ Ordonnancement des sous-activités des spécifications lexicales



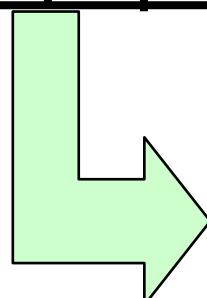
3.3.5) Spécifications lexicales (23)

SL-IHM // Conception du NA

Spécifications lexicales dans les phases USDP

EP	E	C	T

- spécification du vocabulaire terminal
- définition des messages
- création du lexique des objets graphiques
- conception des fenêtres principales et secondaires
- rédiger les tableaux ICAR (mode d'emploi)
- mise à jour du dossier de conception du NA
- Mettre à jour le glossaire



DLx-IHM *Description lexicale de l'IHM*

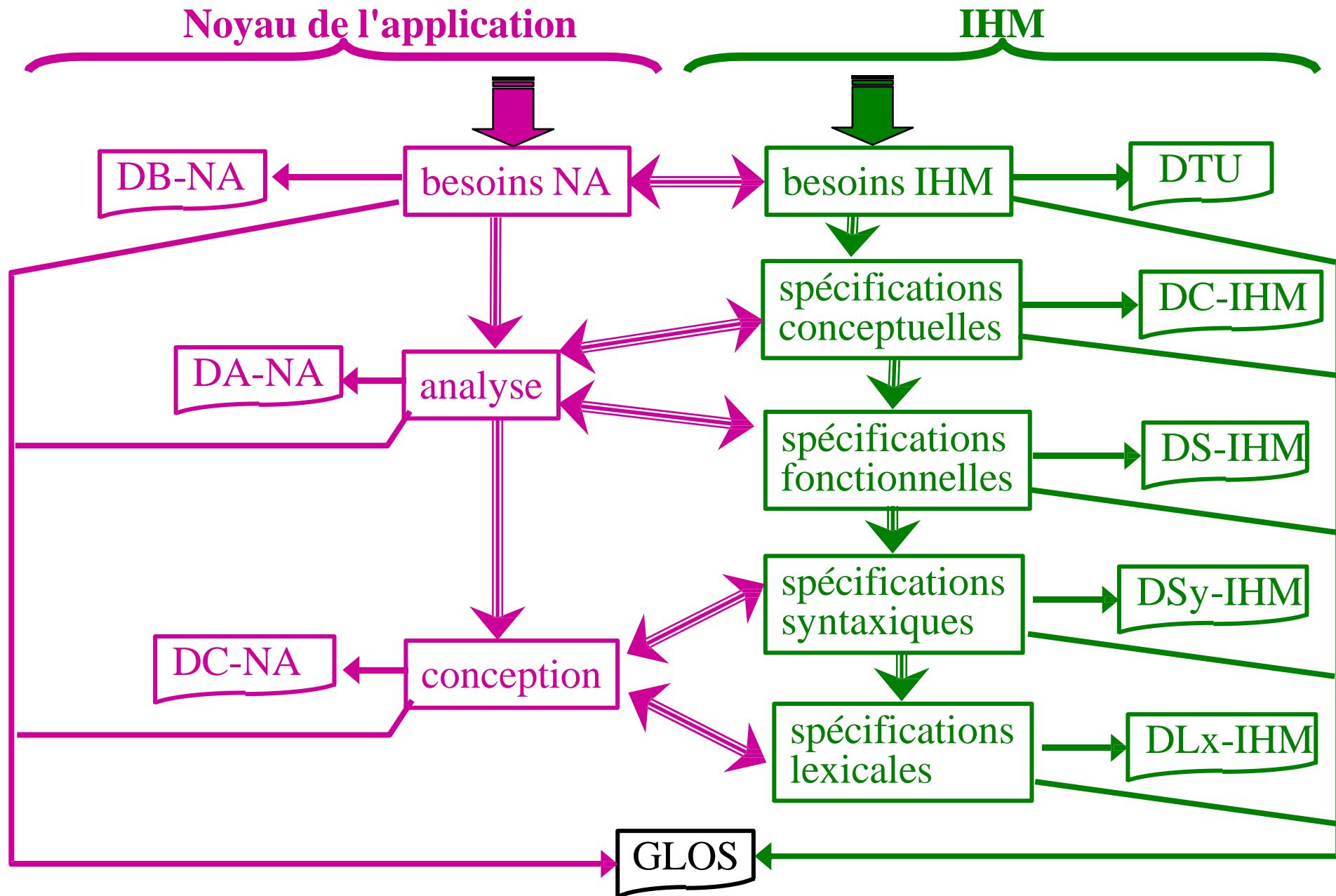
EP: étude préliminaire

E : élaboration

C : Construction

T : transition

3.3.5) Spécifications lexicales (24)



Chapitre 3 : Méthode de conception 00 d'une IHM

- 1) *Présentation de la méthode*
- 2) *Études des besoins pour la CHM*
- 3.3) *Spécifications de l'IHM*

➤ 3.4) Réalisation et tests

✓ 3.4.1) Outils pour la réalisation

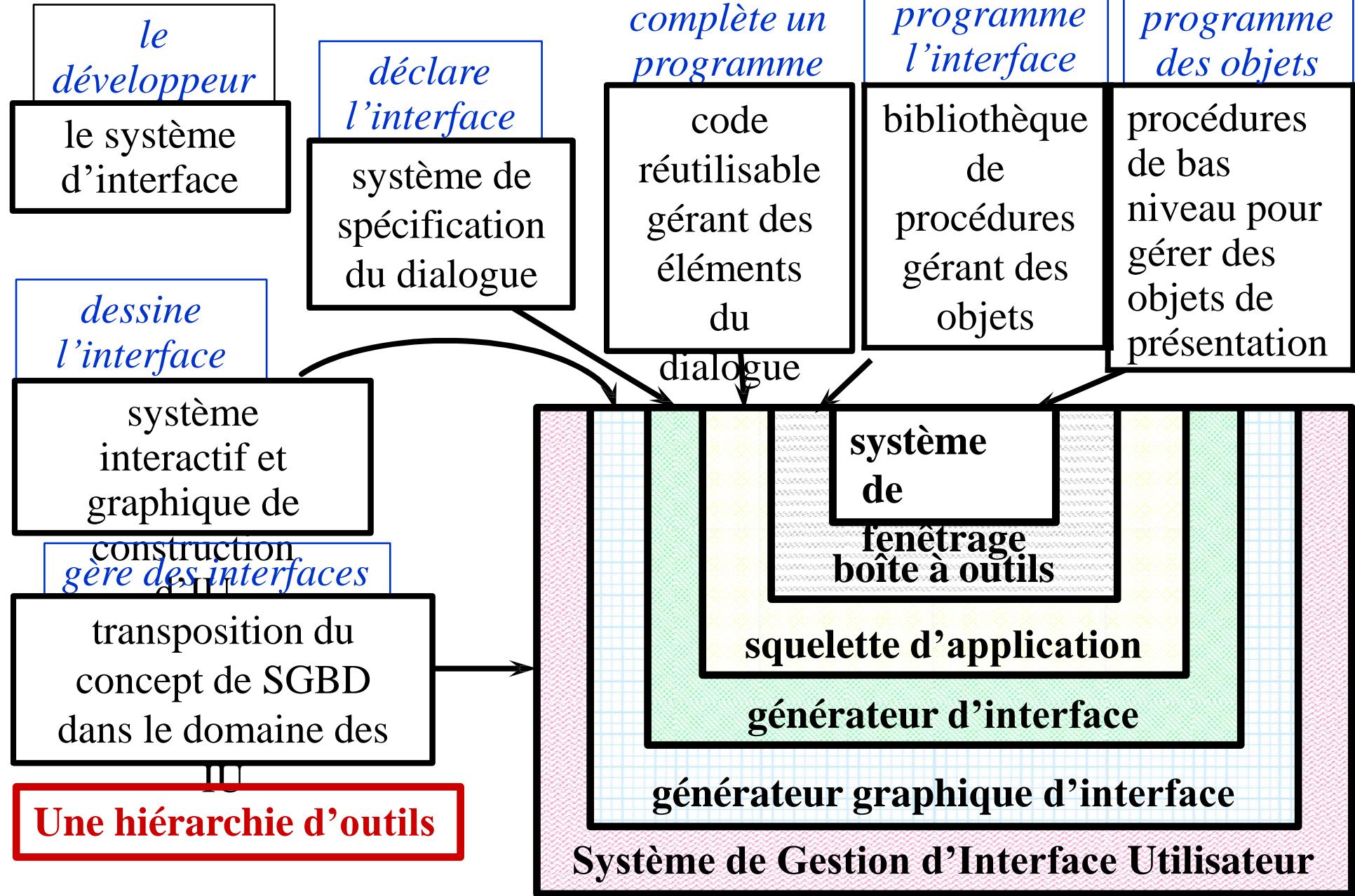
3.4.2) Réalisation et tests

3.4.1) Outils pour la réalisation (1)

- **L'implémentation en // du noyau de l'application et de l'IHM**
- **Les outils pour développer des IHM**
 - boites à outils associées à un langage de programmation et un système d'exploitation avec gestion de fenêtres
 - générateurs d'interfaces (*graphique ou non*)
 - système de gestion d'interface utilisateur (*SGIU*)

3.4.1) Outils ... (2)

Il est exclu de construire complètement une IHM.



3.4.1) Outils pour la réalisation (3)

□ Gestionnaire de fenêtre

gère les fenêtres créer, modifier, détruire
 échanger entre fenêtre

- *exemple : X-Windows*

- bâti sur le modèle client-serveur
- hiérarchie de fenêtres
- des objets du monde X dans ces fenêtres
 - hiérarchie d'héritage (spécialisation)
 - hiérarchie d'assemblage (composition)

- intègre des outils de représentation

- fontes de caractères
- des graphiques élémentaires
- des systèmes graphiques évolués



3.4.1) Outils pour la réalisation (4)

■ Boîte à Outils

❖ Bibliothèque d'objets interactifs et de procédures

- située au dessus du gestionnaire de fenêtre
- offre au développeur une API
- elle ne gère pas le dialogue
 - ✓ mais elle est accompagnée d'un guide de style

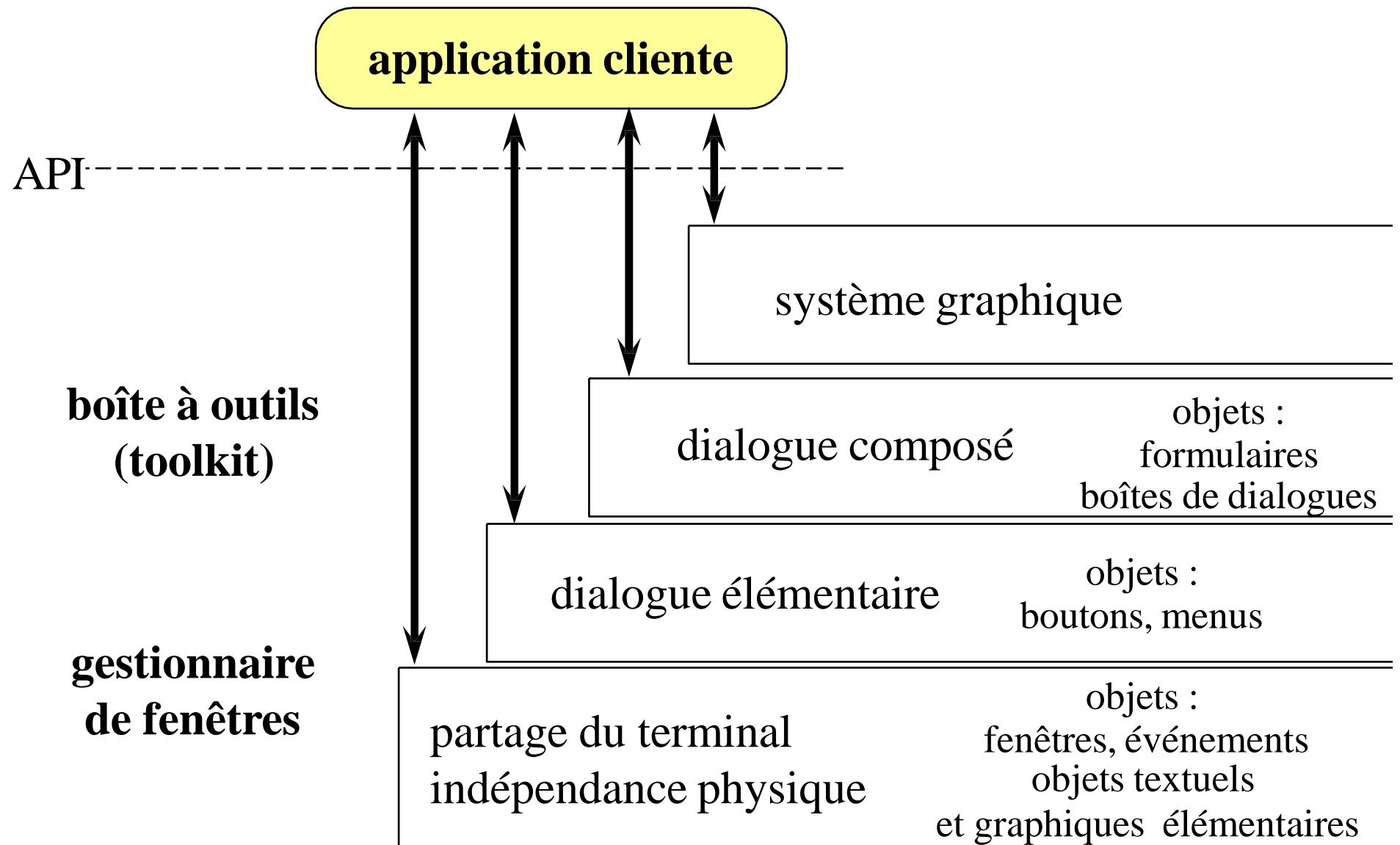
❖ avantages

- niveau d'abstraction pour le développeur
- consistance entre applications
- niveau d'intégration entre applications
- généralement extensible

❖ inconvénients

- difficile à utiliser (*connaissance de l'AOO, C++, JAVA ou ~*)
- duplication de code
- attention à la portabilité

3.4.1) Outils pour la réalisation (5)

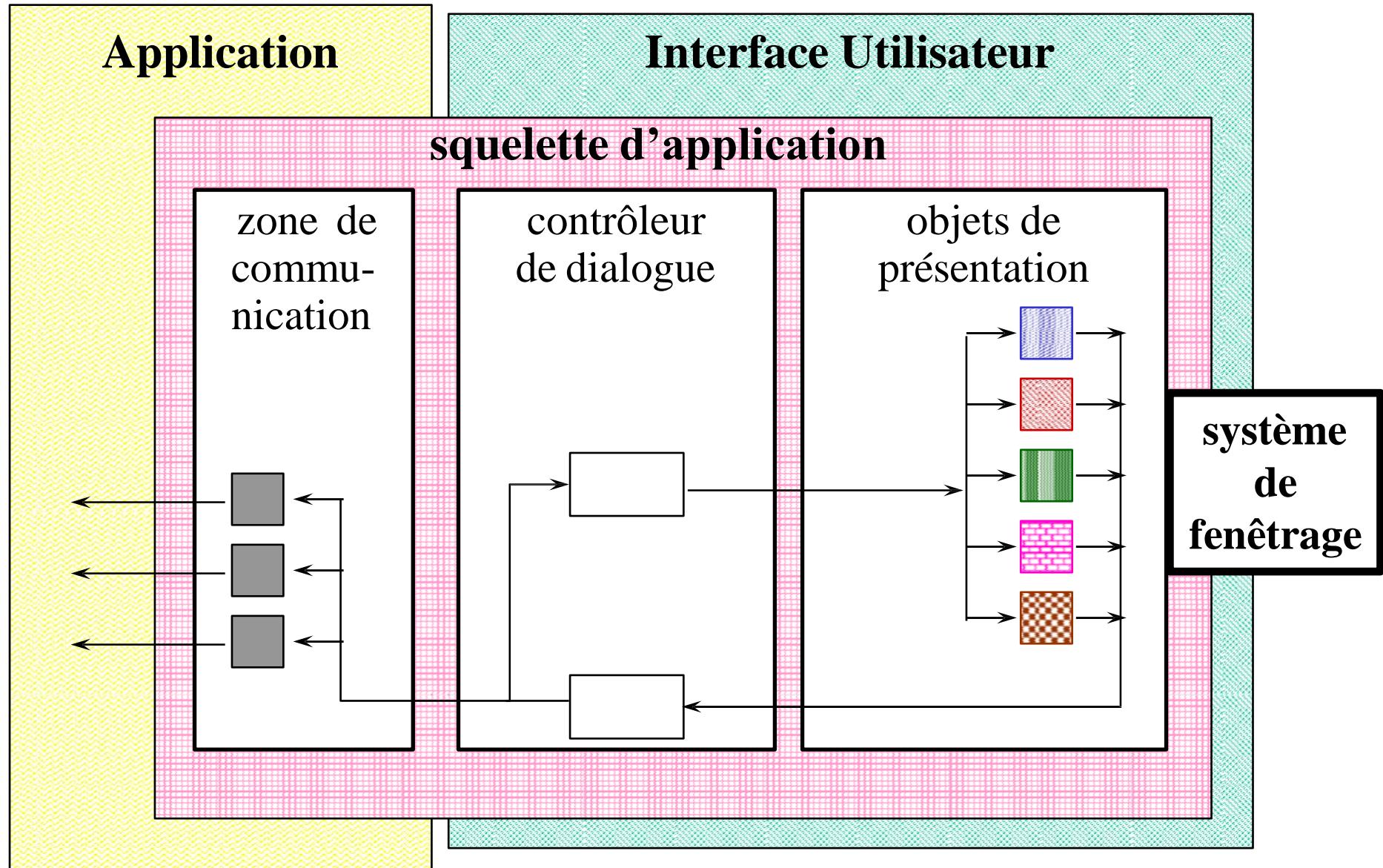


3.4.1) Outils pour la réalisation (6)

□ Squelette d'application

- ❖ **Dans une application** il existe de nombreuses séquences identiques *ex. afficher une fenêtre (contenu + dialogue de base)*
- ❖ **un squelette d'application est une structure d'application**
 - interactive
 - réutilisable
 - adaptable (paramétrable) : il y a des blancs à remplir
 - généralement, noyé dans le générateur d'interface
- ❖ **Un squelette d'application** fournit à l'application une interface de haut niveau qui
 - intègre une part du dialogue
 - est prête à l'emploi
 - permet de réutiliser du code
 - est ouverte vers les LOO
 - permet de personnaliser un style
- ❖ est réservé aux professionnels de la programmation d'IHM

3.4.1) Outils pour la réalisation (7)



3.4.1) Outils pour la réalisation (8)

□ Le générateur d'interface

- **génère automatiquement le code** à partir de spécifications de haut niveau: langage graphique
langage quasi-naturel
- **se divise en deux parties**
 - le sous-système de génération
 - le sous système d'exécution
- **il existe # langages de spécif. de l'interface** fonctionnels
 - déclaratifs
 - à objets
- **on peut les associer à une couche interactive**
 - qui permet de dessiner l'interface
 - qui aide le concepteur dans les spécifications

3.4.1) Outils pour la réalisation (9)

□ Générateur interactif graphiques d'interface (GIGI) la construction d'une interface comporte deux phases

L e dessin de l'interface (réalisé par un graphiste)

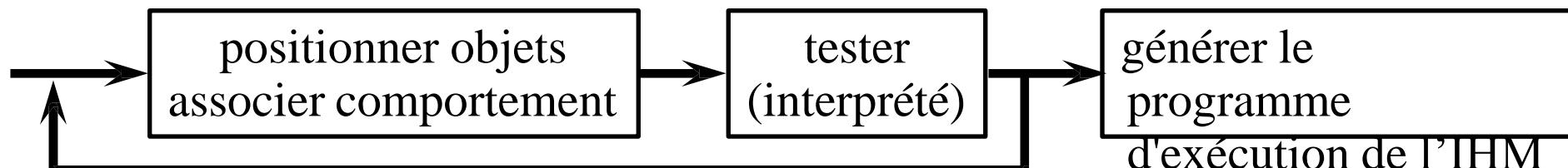
- place les objets sur l'écran (en manipulation directe)
- fixe les attributs (couleur, taille, ...)
- opération WYSIWYG
- les objets sont nommés

La définition du comportement de l'IHM

- affectation de comportements pré-programmés
- programmation d'un nouveau comportement

Ces deux phases peuvent être imbriquées

- dessiner chaque objet et lui affecter son comportement



3.4.1) Outils pour la réalisation (9)

Système de Gestion des Interfaces Utilisateurs

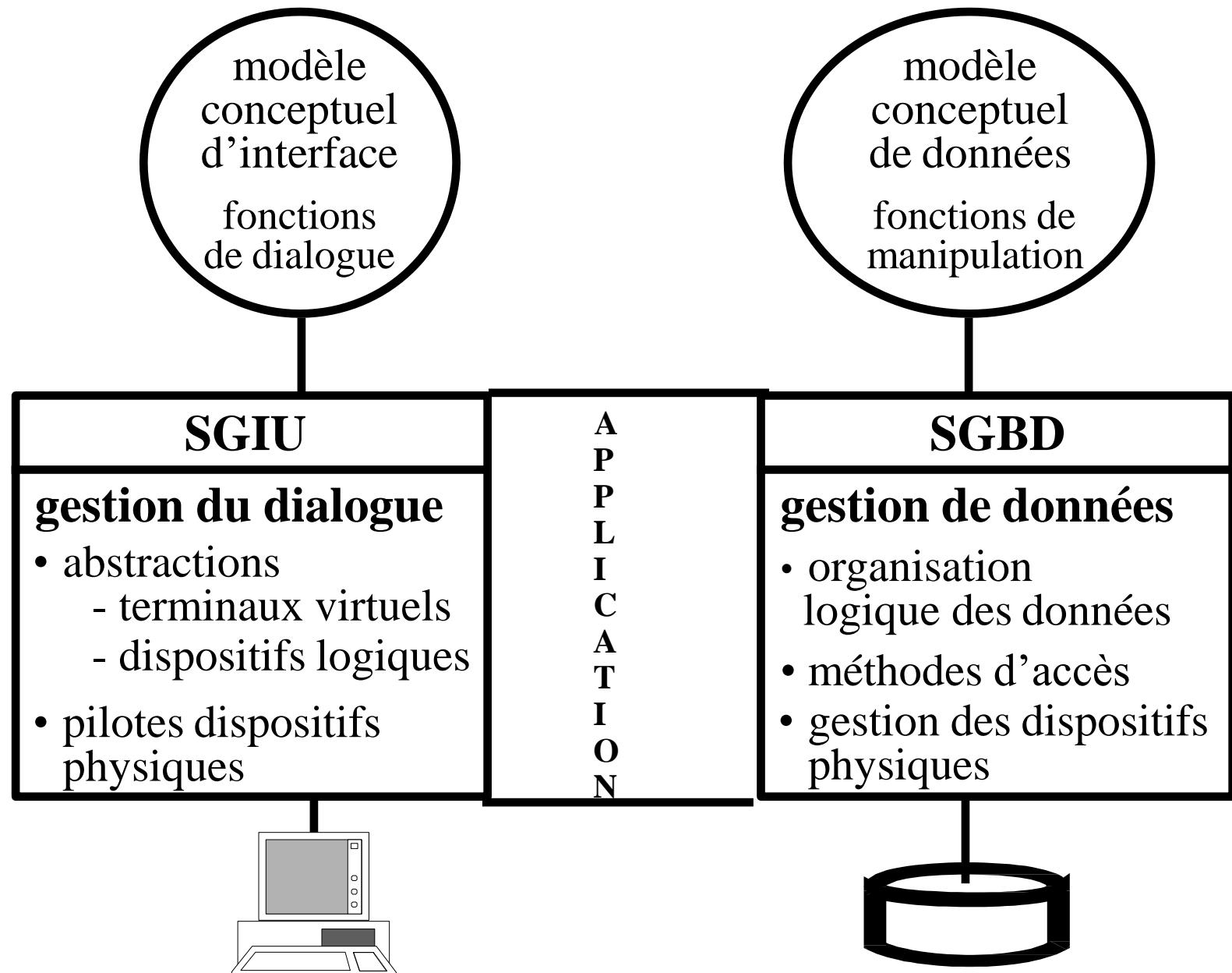
Analogie avec les SGBD

- modèle de données \Leftrightarrow modèle de dialogue
- indépendance (données / application) \Leftrightarrow (dialogue / application)
- partage de données \Leftrightarrow cohérence de l'interface

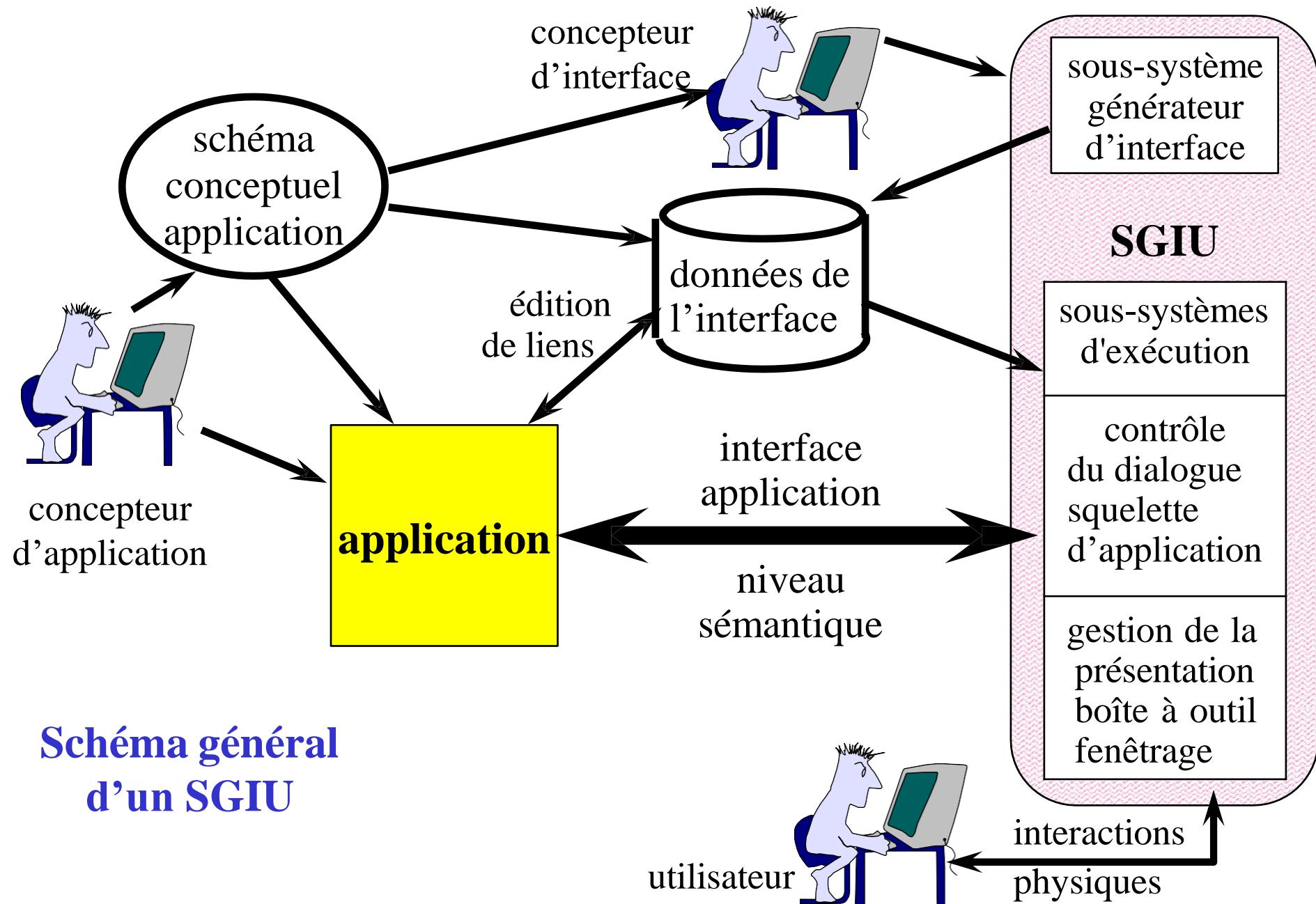
Nombreux problèmes non résolus

- **niveau d'abstraction du protocole d'échange** entre SGIU et application
- **localisation du contrôle du dialogue**
 - si application : risque de blocage en cas de réactions imprévues de l'utilisateur
 - si SGIU asynchronisme est mal pris en compte
ouverture du dialogue + limité
 - si les 2: pb de maintenance

3.4.1) Outils pour la réalisation (10)



3.4.1) Outils pour la réalisation (11)



Chapitre 3 : Méthode de conception 00 d'une IHM

- 1) Présentation de la méthode
- 2) Études des besoins pour la CHM
- 3.3) Spécifications de l'IHM

➤ 3.4) Réalisation et tests

- 3.4.1) Outils pour la réalisation

✓ 3.4.2) Réalisation et tests

3.4.2) Réalisation et tests (1)

- **Réaliser en // les tests unitaires des objets du N.A. et de l'IHM**
- **Deux démarches complémentaires**

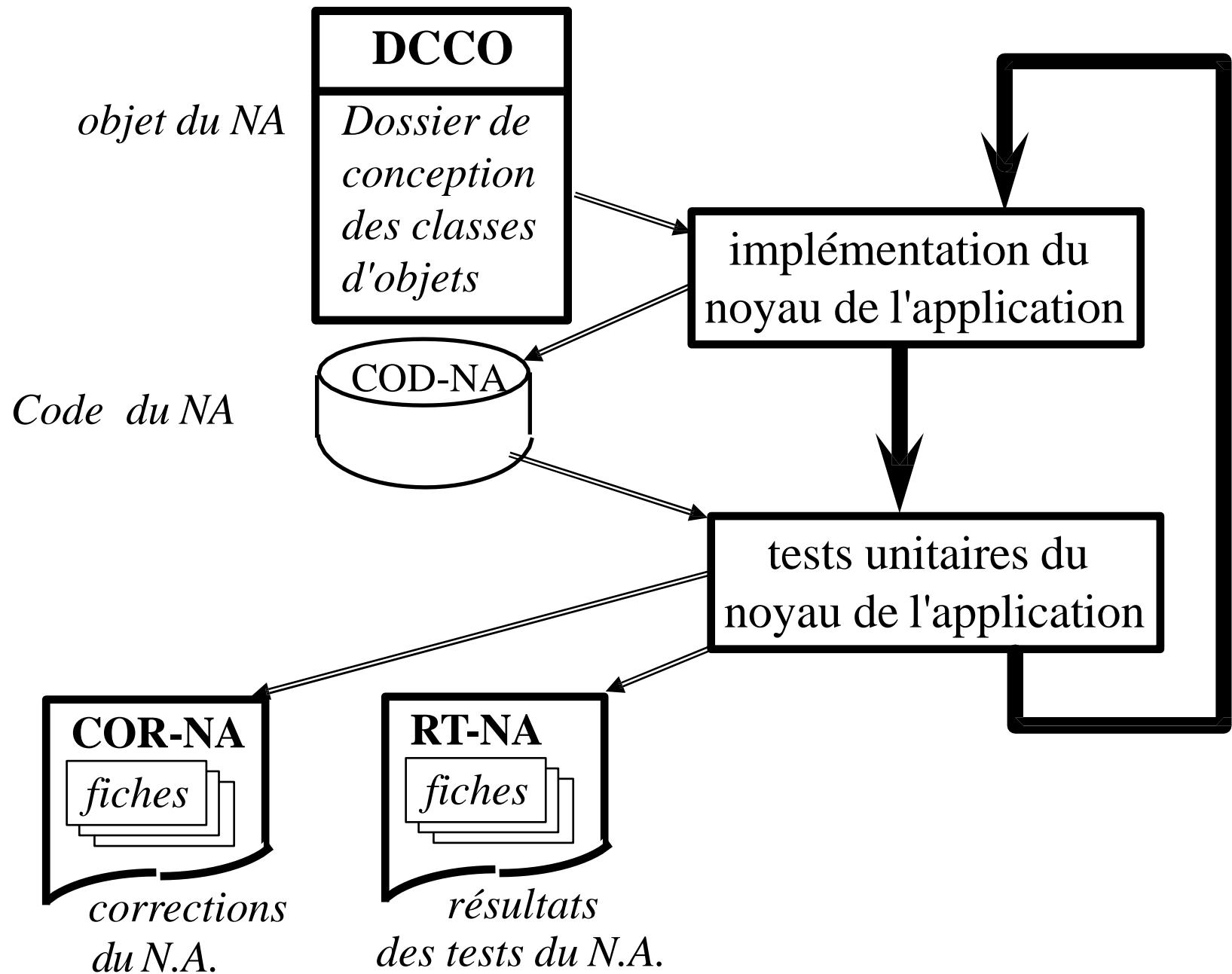
tests en boîte blanche :

- chaque objet est vu et validé par des vérificateurs (# *l'auteur*)
- ils peuvent être effectués très tôt, d'une manière incrémentale

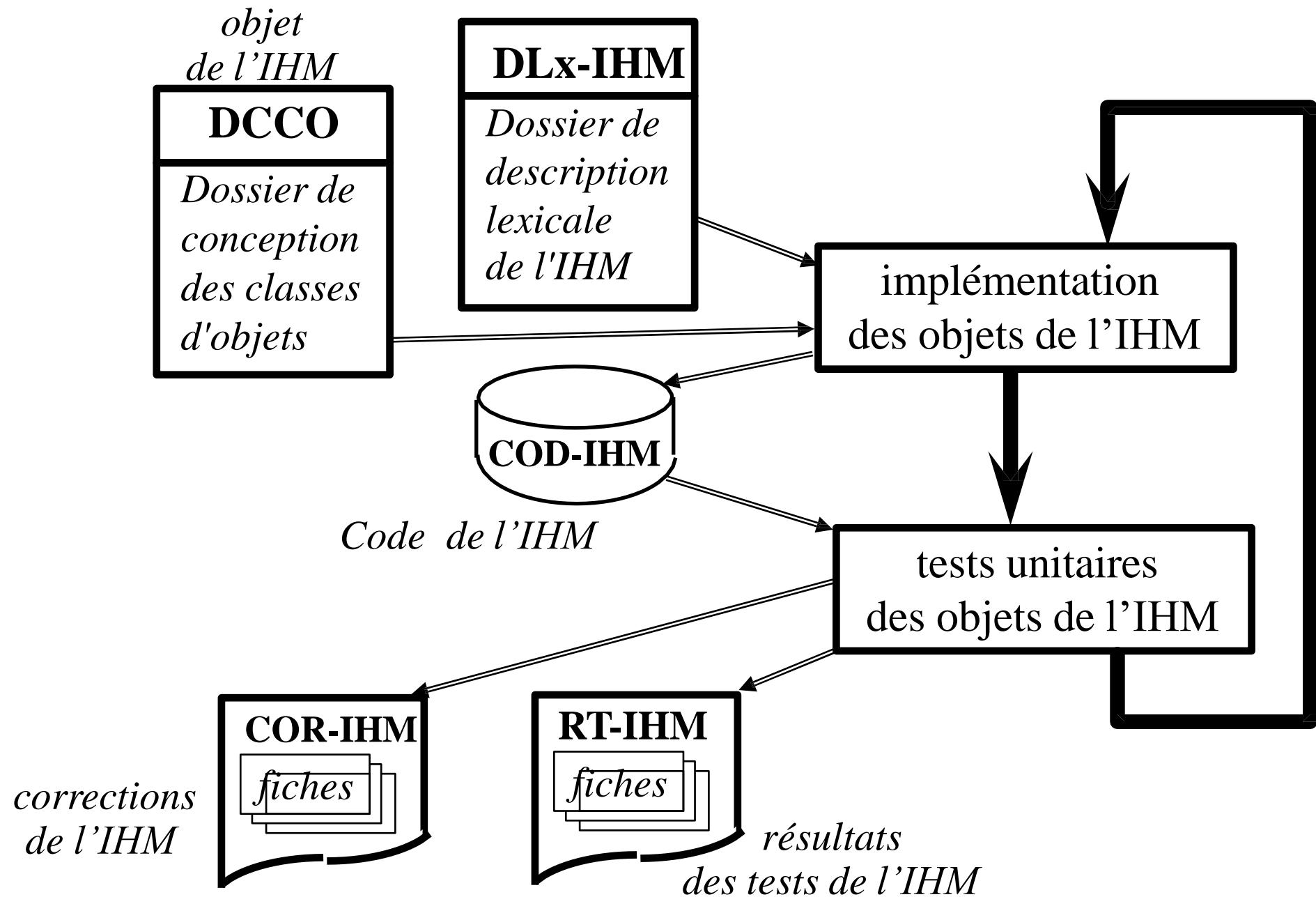
tests en boîte noire :

- on crée un environnement artificiel de test
- ils peuvent aussi être construits progressivement

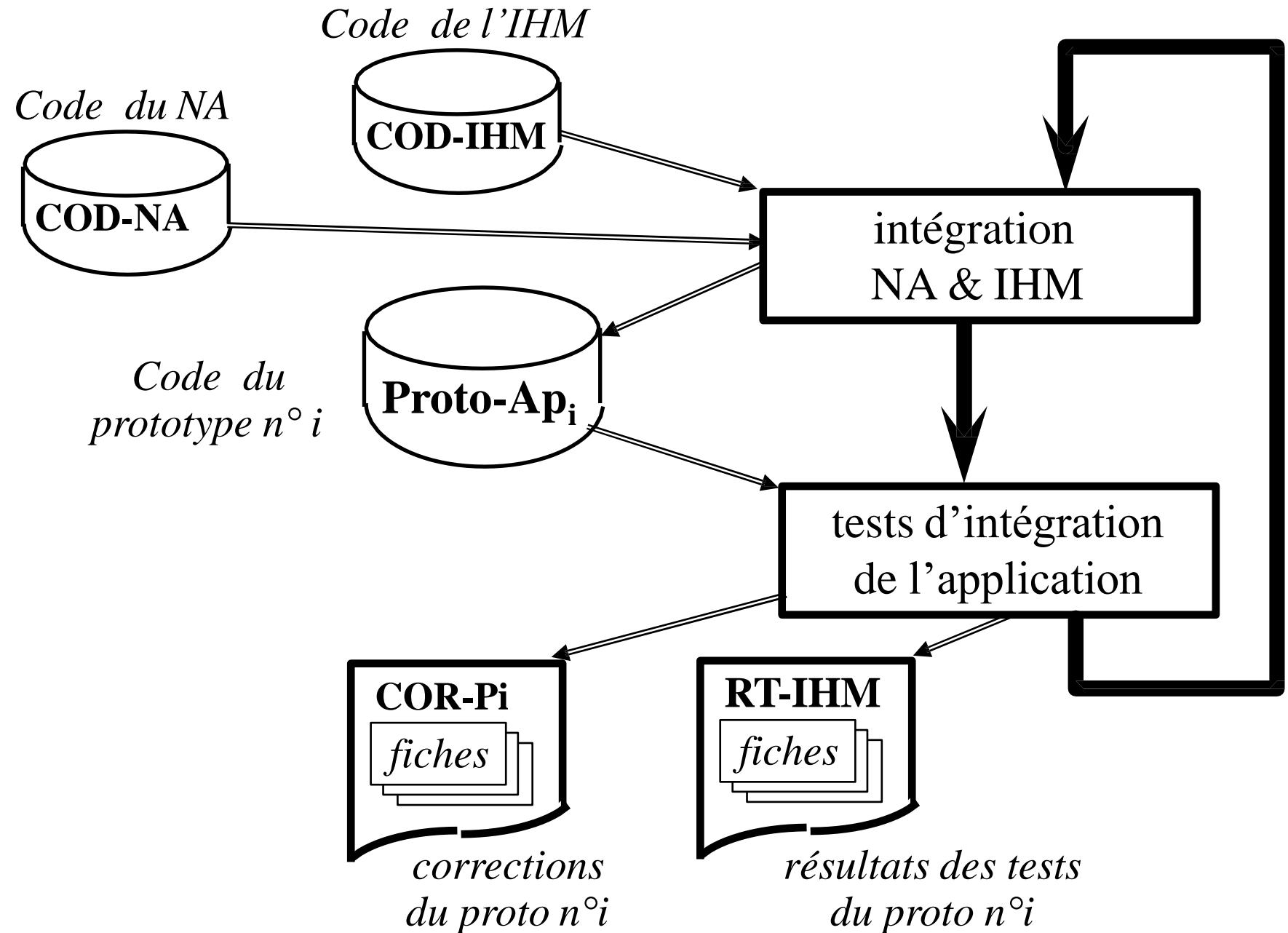
3.4.2) Réalisation et tests (2)



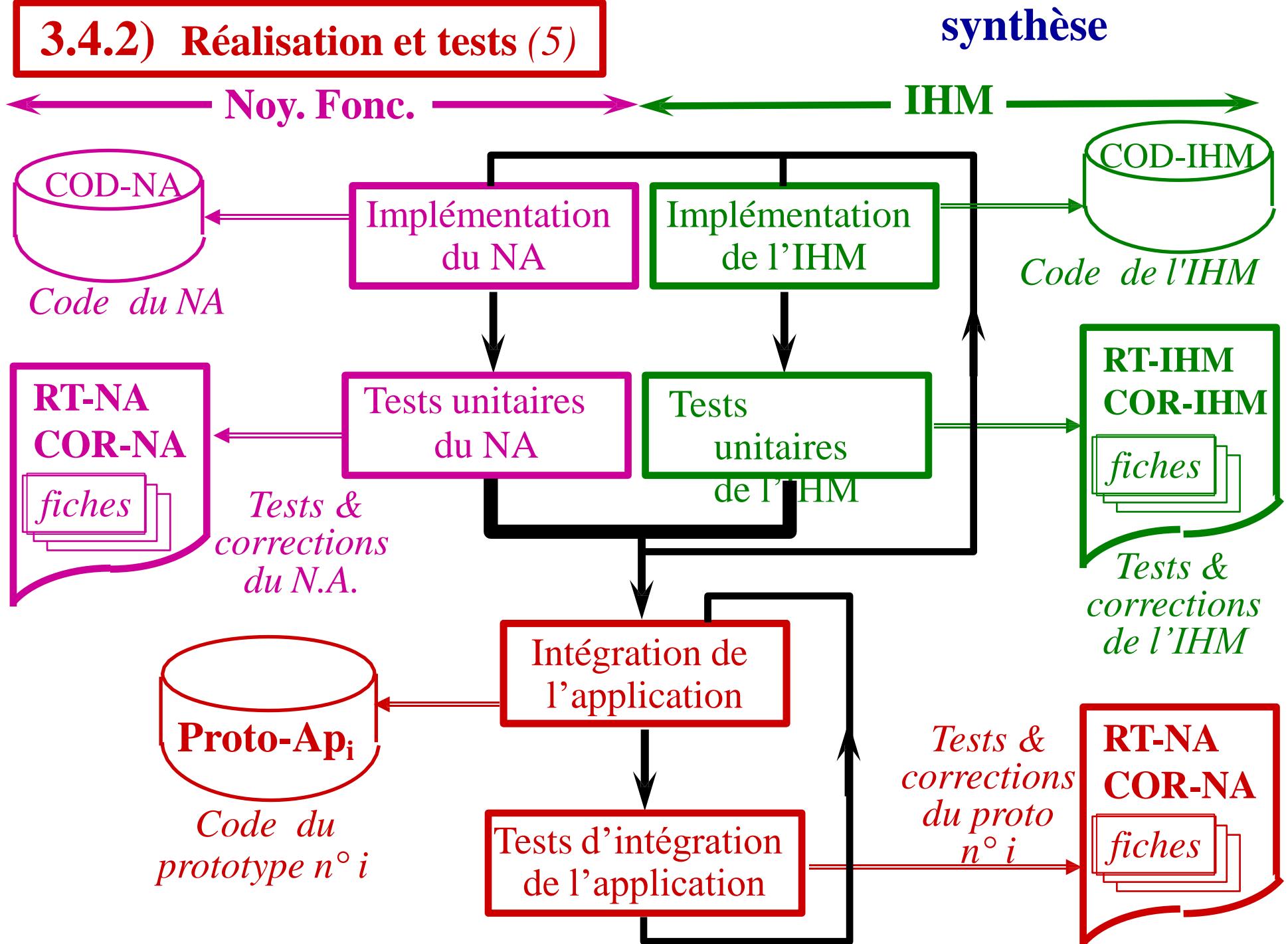
3.4.2) Réalisation et tests (3)



3.4.2) Réalisation et tests (4)



3.4.2) Réalisation et tests (5)



3.4.2) Réalisation et tests (6)

□ Organisation des tests

- **en cours de conception** papier
 maquettes incrémentales
- ✓ **en cours de réalisation** prototypage
 déverminage
- ✓ **avant diffusion** vérification des performances d'usage
 test d'acceptabilité
 recette
- ✓ **après diffusion** enquête et expertise chez les utilisateurs
 création d'un groupe d'utilisateurs
- ✓ **mode d'évaluation**
 - observation directe de l'utilisateur
 - observation via une vidéo (moins contraignante)
 - entretiens avec les utilisateurs (questionnaires)
 - mesures de paramètres grâce à des espions et/ou des capteurs

3.4.2) Réalisation et tests (7)

□ expérimentation contrôlée

✓ expérimentation très lourde

- analyse en profondeur du problème
- doit être réalisée par des spécialistes externes
- justifiée pour des produits très grande diffusion
stratégiques (*militaires, santé*)

□ tests d'usage (cf Spécifications de Performances d'Usage):

✓ Ils sont réalisés sur un prototype très avancé

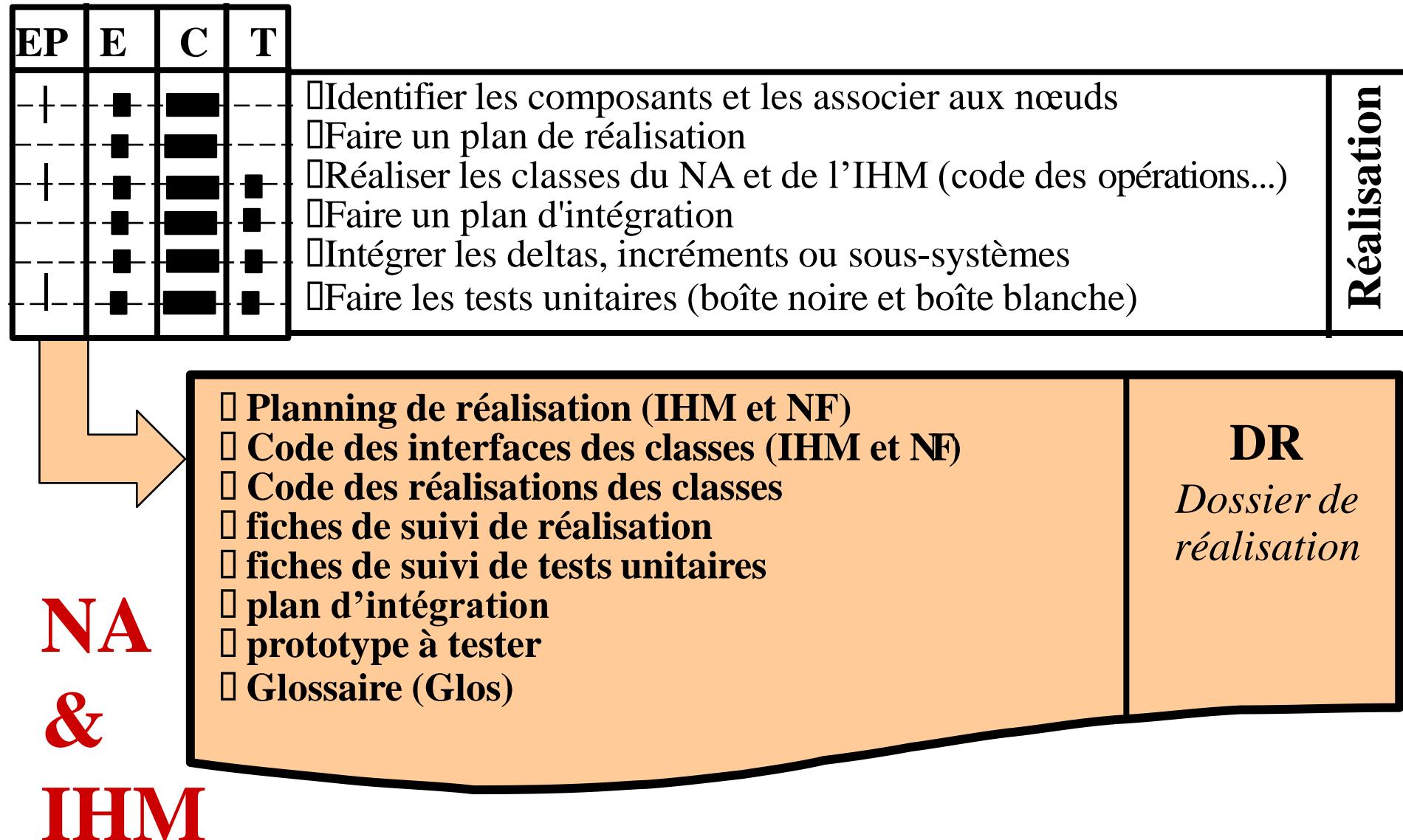
✓ ils permettent d'analyser le temps d'apprentissage les temps de réponse les types d'erreur les + fréquents l'usage de l'aide et de la documentation

✓ ils sont efficaces si l'échantillon d'utilisateur est significatif

- les tâches choisies sont typiques
- les résultats sont analysés rigoureusement
- des alternatives sont testées

✓ ils peuvent être itératifs (versions successives)

3.4.2) Réalisation et tests (8)



3.4.2) Réalisation et tests (9)

Réalisation IHM et NA

EP	E	C	T
■	■	■	
■	■	■	
■	■	■	■
■	■	■	
■	■	■	
■	■	■	
■	■	■	
■	■	■	
■	■	■	

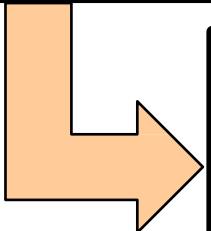
- Rédiger un plan de test (intégration, système, ...)
- Concevoir les tests (intégration, système, non-régression)
- Structurer les procédures de test, automatiser
- Réaliser les tests d'intégration
- Réaliser les tests du système
- Évaluer les tests
- Installer sur le site, faire migrer les données..
- Gérer les retours, adapter et corriger le produit

Tests

NA

&

IHM

- 
- Plan de test : stratégie, effort..
 - Cas de test, procédures, tests automatiques...
 - Document de suivi des défauts
 - Manuel utilisateur, documents de formation...
 - Manuel de maintenance,
 - Version bêta
 - Glossaire (Glos)

DT
Dossier de Tests

3.4.2) Réalisation et tests (10)

□ Lien avec le prototypage incrémental

✓ construction incrémental du prototype (jetables ou non)

- P1 = les écrans seuls
- P2 = P1 + entrées (*clavier - souris*)
- P3 = P2 + fonctions limitées
- P4 = P3 + fonctions complètes, performances limitées
- P5 = P4 + performance normal (**prototype final**)

convient plus particulièrement aux projets de taille réduite

peut être adaptée à la méthode OO pour des gros projets

la conception / réalisation de l'IHM doit être en avance sur celle du noyau de l'application