

Interface Homme Machine

1. Introduction
2. Définitions
3. Critères ergonomiques
4. Les étapes du processus de développement et IHM
- 5. Modèles de tâches**
6. Architecture logicielle des systèmes interactifs
7. Les formalismes de validation des DHMs (Dialogues Homme Machine)

5. Modèles de tâches

Les modèles de tâches sont des **outils de spécification** de l'interaction qui s'appliquent aux différents niveaux d'abstraction :

- ✓ des tâches élémentaires : représentent des suites d'actions physiques (manipulation directe).
- ✓ aux tâches composées ou abstraites : décomposition de tâches en sous-tâches et relations temporelles entre tâches.

5.1 CTT : Concur Task Trees

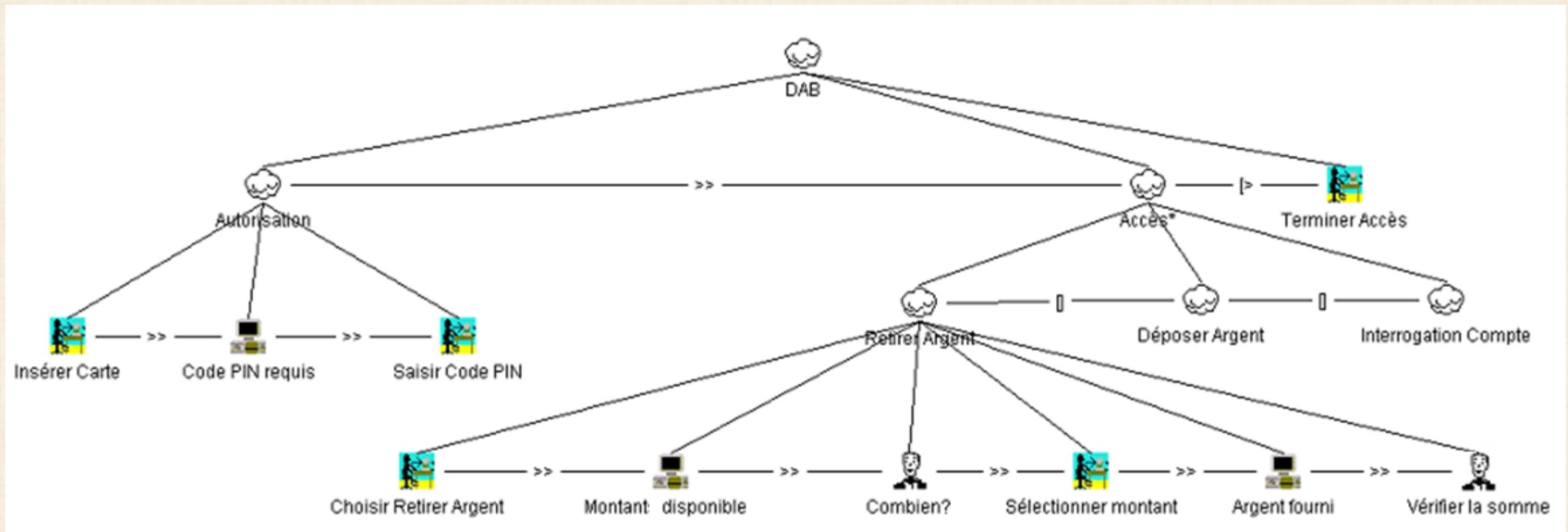
CTT utilise une représentation graphique et propose un outil constituant un environnement d'édition, de simulation, et génération de scénarii de tâches, appelé CTTE (CTTEnvironment) qui est utilisé largement dans le monde académique <http://giove.isti.cnr.it/lab/research/CTTE/home>

5.1 CTT : Concur Task Trees (Cont.)

Exemple

La figure en bas représente le modèle de tâches d'un **distributeur automatique de billet (DAB)**.

Dans ce modèle l'utilisateur doit **insérer sa carte**, ensuite (**>>**) **saisir son code PIN** pour avoir l'autorisation d'accéder à son compte. La tâche **Accès** peut être effectuée plusieurs fois (*) et désactivée ([>]) à tout moment par la tâche **Terminer Accès**. L'utilisateur choisit ([]) ensuite de **retirer de l'argent** (Retirer Argent), de **déposer de l'argent** (Déposer Argent) ou bien de **consulter son compte** (Interrogation Compte).



5.1 CTT : Concur Task Trees (Cont.)

Le modèle formel

ConcurTaskTrees peut être formellement décrit à l'aide d'une méthode dite B* événementielle donnant ainsi :

- ✓ La possibilité d'exprimer à la fois la conception de l'IHM et
- ✓ L'expression des tâches utilisateurs.
- ✓ Cette dernière permet de décrire formellement la sémantique de CTT suivant sa grammaire BNF (vue au cours 1).

Task ::=	$Task >> Task$	-- Activation
	$Task \square Task$	-- Choix
	$Task_{At}$	-- Tâche atomique
	$Task \models Task$	-- Ordre indépendant
	$[Task]$	-- Tâche optionnelle
	$Task[> Task$	-- Désactivation
	$Task^*[> Task$	-- Désactivation d'une tâche itérative
	$Task > Task$	-- Interruption
	$Task Task$	-- Concurrence
	$Task^N$	-- Tâche itérative finie

*Le modèle B classique est une méthode formelle qui s'appuie sur des bases mathématiques (logique de premier ordre et théorie des ensembles).

5.1 CTT : Concur Task Trees (Cont. Le modèle formel)

- ✓ Les règles de cette grammaire ne présentent que le déroulement temporel des tâches qui est défini par la sémantique de Lotos.
- ✓ À cela s'ajoutent aussi les informations relatives à la manipulation des objets par les tâches,
- ✓ il s'agit plus particulièrement de la formalisation des catégories (tâche abstraite, interactive, etc), des pré-conditions et des post-conditions.
- ✓ Ainsi l'approche B évènementielle permet de décrire le modèle CTT en se basant sur l'algèbre de processus** sous-jacente à CTT.
- ✓ Cette formalisation permet le codage, par raffinement des arbres de tâches issus de CTT.

** L'Algèbre de processus est une famille de langages formels fournissant des outils formels permettant de caractériser les interactions entre processus au sein d'un système concurrent ou distribué.

5.1 CTT : Concur Task Trees (Cont. Modèle formel)

- ✓ Chaque opérateur de CTT est représenté par un raffinement.
- ✓ Ainsi une règle BNF de la forme **T0 ::= T1 op T2** est décomposée en deux machines :
 - la première contient l'événement **T0** et
 - la seconde, raffinant la première, contient les événements **T0, T1 et T2** et effectue la décomposition de **T0** en **T1 op T2**.

la méthode B possède des outils de preuves automatiques commercialisés. Les outils les plus répandus sont **Atelier-B*** et **B-Toolkit****.

*ClearSy : <http://www.clearsy.com>

**B-Core (UK) Limited : <http://www.b-core.com>

5.1 CTT : Concur Task Trees (Cont. Modèle formel)

Exemple d'illustration : l'addition

La somme **Sum** de deux nombres naturels **aa** et **bb** illustre le fonctionnement des règles de traduction du langage CTT en B événementiel.

Ci-dessous un raffinement permettant de calculer la somme des deux nombres naturels.

Le modèle B événementiel **Task_SumT0** représente une instance de la tâche **T0**.

```
MODEL Task_SumT0
INVARIANT
  Sum ∈ NAT ∧ aa ∈ NAT ∧ bb ∈ NAT
INITIALISATION
  Sum :∈ NAT || aa :∈ NAT || bb :∈ NAT
EVENTS
  Evt0 =
BEGIN
  Sum := aa + bb
END ;
```

5.1 CTT : Concur Task Trees (Cont. Modèle formel)

Exemple de raffinement pour l'opération d'**addition** dans B événementiel.

```

REFINEMENT RefEnablingTaskT0
REFINE Task_SumT0
INVARIANT
  (RSum + AA + BB) = (aa + bb)  $\wedge \dots$ 
VARIANT
  (AA + BB)
ASSERTIONS
  (AA = 0  $\wedge$  BB = 0  $\wedge$  aa + bb  $\in$  NAT)  $\vee \dots$ 
INITIALISATION
  RSum := 0  $\parallel$  Sum : $\in$  NAT  $\parallel$  aa, AA : $\in$  (aa  $\in$  NAT  $\wedge$  AA = aa)  $\parallel \dots$ 

EVENTS
  Evt1 =           Evt2 =           Evt0 =
SELECT           SELECT           SELECT
  AA  $\neq$  0  $\wedge$  BB  $\neq$  0  $\wedge \dots$    AA = 0  $\wedge$  BB  $\neq$  0  $\wedge \dots$    AA = 0  $\wedge$ 
THEN             THEN             THEN
  RSum := RSum + AA    RSum := RSum + BB    Sum := RSum
   $\parallel$  AA := 0         $\parallel$  BB := 0
END;          END;          END;

```

5.1 CTT : Concur Task Trees (Cont. Modèle formel)

- ✓ La clause **INITIALISATION** permet d'initialiser l'ensemble des variables de telle façon que **Sum**, **aa** et **bb** choisissent des valeurs parmi des nombres naturels.
- ✓ Ce modèle contient l'événement **Evt0** où sa garde est vraie et son corps permet le calcul de la somme.
- ✓ Pour les différents raffinements de cette illustration les nouvelles variables: **RSum**, **AA** et **BB** sont utilisées.
- ✓ Elles correspondent respectivement aux variables raffinées de celles abstraites **Sum**, **aa** et **bb**.
- ✓ Ces variables sont liées entre elles par l'invariant de collage (**RSum+AA+BB = aa+bb**) qui assure la correspondance entre les deux niveaux de modélisation et par conséquent au codage des opérateurs CTT.
- ✓ Ainsi l'entier Naturel **AA + BB** déterminera le variant. Il diminuera jusqu'à **0** pour déclencher l'événement raffiné de l'abstraction.

5.1 CTT : Concur Task Trees (Cont. Modèle formel)

- ✓ La somme de **aa** et **bb** soit réalisée de manière séquentielle.
- ✓ Tout d'abord la variable **aa** est affectée à **RSum** par l'événement **Evt1** et ensuite la variable **bb** est ajoutée à **RSum** par l'événement **Evt2**.
- ✓ Ces deux événements du raffinement travaillent pour l'événement **Evt0** dont le rôle est de modifier **Sum** et de raffiner **Evt0** de l'abstraction.

EVENTS		
<i>Evt₁</i> =	<i>Evt₂</i> =	<i>Evt₀</i> =
SELECT	SELECT	SELECT
$AA \neq 0 \wedge BB \neq 0 \wedge \dots$	$AA = 0 \wedge BB \neq 0 \wedge \dots$	$AA = 0 \wedge$
THEN	THEN	$BB = 0 \wedge \dots$
$RSum := RSum + AA$	$RSum := RSum + BB$	THEN
$\parallel AA := 0$	$\parallel BB := 0$	$Sum := RSum$
END;	END;	END;

- ✓ La clause **ASSERTIONS** assure que les nouveaux événements sont déclenchés et que le variant (modélisé par les variables de l'**addition**) garantit le déclenchement en séquence.

5.2 Le formalisme UAN: User Action Notation

UAN signifie la notation des actions de l'utilisateur, ce formalisme est prévu pour être écrit principalement par quelqu'un qui modélise les composants interactifs d'une interface c.-à-d. un ergonome, et pour être lu par tous les développeurs, en particulier ceux qui implémentent le noyau fonctionnel de l'application.

L'UAN est un mécanisme utilisé pour décrire la conception d'une IHM qui doit être : Précise, concise (brève), non ambiguë et détaillée.

Description de l'UAN:

- L'interaction (ensemble de tâches utilisateurs) est modélisée par des notations sous forme de tables.
- L'UAN modélise le comportement User-Interface comme ils accomplissent une tâche ensemble.
- Basé sur la tâche comme élément primaire d'analyse.
- L'interface complète est vue comme une hiérarchie de tâches non ordonnées.
- Le niveau le plus bas d'une description UAN représente :
 - **L'action utilisateur** (User Actions)
 - **La réaction du système** (Interface FeedBack)
 - **Le changement d'état** (Interface State)

5.2 Le formalisme UAN: User Action Notation (Cont.)

À tous les niveaux, les actions et les tâches utilisateurs sont contraintes par les relations temporelles suivantes: *Ordonnancement*, *imbrication* et *simultanéité* (tâches parallèles) et complétées par des scénarios (séquence d'images d'écrans) plus des diagrammes de transitions d'états.

Syntaxe UAN: les tâches sont décrites sous forme tabulaire comme suit:

Task: <Name>	User Actions	Interface Feedback	Interface State

5.2 Le formalisme UAN: User Action Notation (Cont.)

Opérateurs d'UAN:

1) Les actions

Les symboles modélisant les actions de l'user

Action	Signification
\sim	déplace le curseur
$[X]$	contexte de l'objet X
$\sim [X]$	déplace le curseur sur l'objet X
$\sim [x, y]$	déplace le curseur en un point extérieur à tout objet
$\sim [x, y \text{ in } A]$	déplace le curseur en x,y dans A
$\sim [X \text{ in } Y]$	déplace le curseur en X dans Y
$[X] \sim$	déplace le curseur hors de X
\vee	appuie
\wedge	relâche
X^\vee	appuie sur un bouton, une touche ou l'objet X
X^\wedge	relâche un bouton, une touche ou l'objet X
$X^{\vee\wedge}$	idiome pour le clic sur un bouton, une touche ou l'objet X
$X "abc"$	entrer la chaîne abc via le media X
$X(xyz)$	entrer la valeur de la variable xyz via le media X
$()$	mécanisme de groupe
$*$	fermeture itérative, la tâche est exécutée 0 ou plusieurs fois
$+$	la tâche est exécutée 1 ou plusieurs fois
$\{ \}$	tâche optionnelle (exécutée 0 ou plusieurs fois)
$A ; B$	séquence ; exécution de A, puis de B
OR	disjonction, choix de tâche
$\&$	ordre relatif des tâches indifférent
\Leftrightarrow	interchangeable
\parallel	concurrente
$;$	symbole d'interruption de tâche
\forall	pour tout
$:$	séparateur entre condition et action ou sortie

5.2 Le formalisme UAN: User Action Notation

2) Le Feed-Back (La réaction de l'interface)

Les symboles modélisant les réactions du système

D'autres opérateurs:

file_icon': Icône sélectionnée

\neg : Non logique.

$=$: Égalité.

$*$: Différent.

Sortie	Signification
!	objet mis en valeur
-!	objet plus mis en valeur
!!	idem !, mise en valeur différente
!-	clignotement
$(!-!)^n$	clignotement n fois
@ x,y	au point x, y
@ X	à l'objet X
display(X)	montre objet X
erase(X)	efface l'objet X
X >~	l'objet X suit le curseur
X >>~	objet X s'agrandit pour suivre le curseur
outline(X)	lignes extérieures de l'objet X

5.2 Le formalisme UAN: User Action Notation

Exemples: (seule la partie des actions est considérée)

1) *Sélectionner un fichier*

Les Actions de l'utilisateur:

La tâche '*Sélectionner un fichier*' peut être décrite informellement par:

- (1) déplacer le curseur vers l'icône du fichier;
- (2) presser et immédiatement relâcher le bouton de la souris.

La partie **action de l'user** selon le formalisme UAN est la suivante:

- (1) $\sim[\text{file_icon}]$
- (2) $M^V \wedge$

La description complète de la tâche '*Sélectionner un fichier*'

Task: Select file		Interface State
User Actions	Interface Feedback	
$\sim[\text{file_icon}] M_V$	$\text{file_icon} !$ $\nexists \text{file_icon} * \text{file_icon}' : \text{file_icon}-!$	$\text{selected} = \text{file}$
M^V		

5.2 Le formalisme UAN: User Action Notation

2) *Déplacer un fichier*

Et la description informelle pour la tâche '**Déplacer un fichier**', est la suivante :

- (1) Déplacer le curseur vers le fichier. Presser et maintenir le bouton de la souris en bas. Presser le bouton de la souris permet de sélectionner le fichier, indiqué par la mise en valeur de son icône.
- (2) Avec le bouton en bas, déplacer le curseur. L'aperçu du fichier suit le curseur comme nous le déplacerons dans l'écran.
- (3) Relâcher le bouton. L'icône du fichier est maintenant déplacée à la place où nous relâchions le bouton.

Les actions utilisateur de la description UAN correspondante est comme suit:

- (1) $\sim [\text{file_icon}] M^V$
- (2) $\sim [x, y]^* \sim [x', y']$
- (3) M^A

La description complète de la tâche '*Déplacer un fichier*' (Voir TD, Série 3)

5.3 Les modèles HTA et MAD

Les modèles MAD et HTA sont similaires dans leur décomposition hiérarchique mais le modèle MAD se diffère de l'HTA par :

- ✓ la description de chaque nœud de l'arbre de tâches
- ✓ qui implique une quantité importante d'information à fournir pour chaque but (ou tâche) identifié dans la hiérarchie.

1) HTA (Hierarchical Task Analysis) :

- ✓ Le modèle HTA est l'une des premières notations utilisées pour la description des activités de l'utilisateur.
- ✓ Sa capacité d'abstraction est réalisée par la décomposition hiérarchique de tâches en sous tâches,
- ✓ HTA admet une description graphique (et une autre textuelle).

- ✓ En HTA, l'ordonnancement de tâches et sous tâches est connu sous le nom de « **plan** ».
- ✓ Ce plan, requis pour tout nœud possédant des sous-tâches, il montre comment il faut utiliser lesdites sous-tâches pour réaliser le but.

1) HTA (Cont.)

Les opérations disponibles pour les plans d'une HTA sont:

Concept	Notation	Commentaires
Sous-tâches d'une tâche	<entier à partir de 1>	Les sous-tâches servant à redéfinir une tâche sont numérotées séquentiellement à partir de « 1 » en commençant par celle de gauche.
Exécution séquentielle	1 > 2 > 3 > 4	Toutes les opérations doivent être exécutées dans l'ordre indiqué.
Exécution non séquentielle	1 / 2 / 3 / 4	Toutes les opérations doivent être exécutées sans égard à l'ordre indiqué.
Exécution libre	1 : 2 : 3 : 4	Un certain nombre d'opérations (mais au moins une) doivent être exécutées, peu importe l'ordre.
Exécution d'un choix	1 2 3 4	Une seule des opérations mentionnées doit être exécutée.
Exécution concurrente	1 + 2 + 3 + 4	Toutes les opérations doivent être exécutées concurremment. En AHT, toutefois, le mot « concurremment » peut aussi bien désigner des opérations à faire en même temps par plus d'une personne (en simultané) que des opérations à faire en temps partagé par une seule personne (en parallèle).
Interdiction	¬ <opération>	Ne pas faire l'opération mentionnée.
Exécution conditionnelle	si X alors 1 sinon 2	Selon que la condition « X » est réalisée ou non, on choisit la sous-tâche 1 ou la sous-tâche 2.
Instructions textuelles	« Facultatif », « Au besoin », « Quand vide », etc.	Des instructions en langage naturel peuvent toujours être ajoutées pour spécifier au mieux les détails du plan.
Regroupement d'instructions pour ciblage précis.	(...)	Lorsque nécessaire, il est possible de regrouper certaines instructions pour y appliquer des conditions particulières ou pour modifier localement la séquence d'exécution.

1) HTA (Cont.)

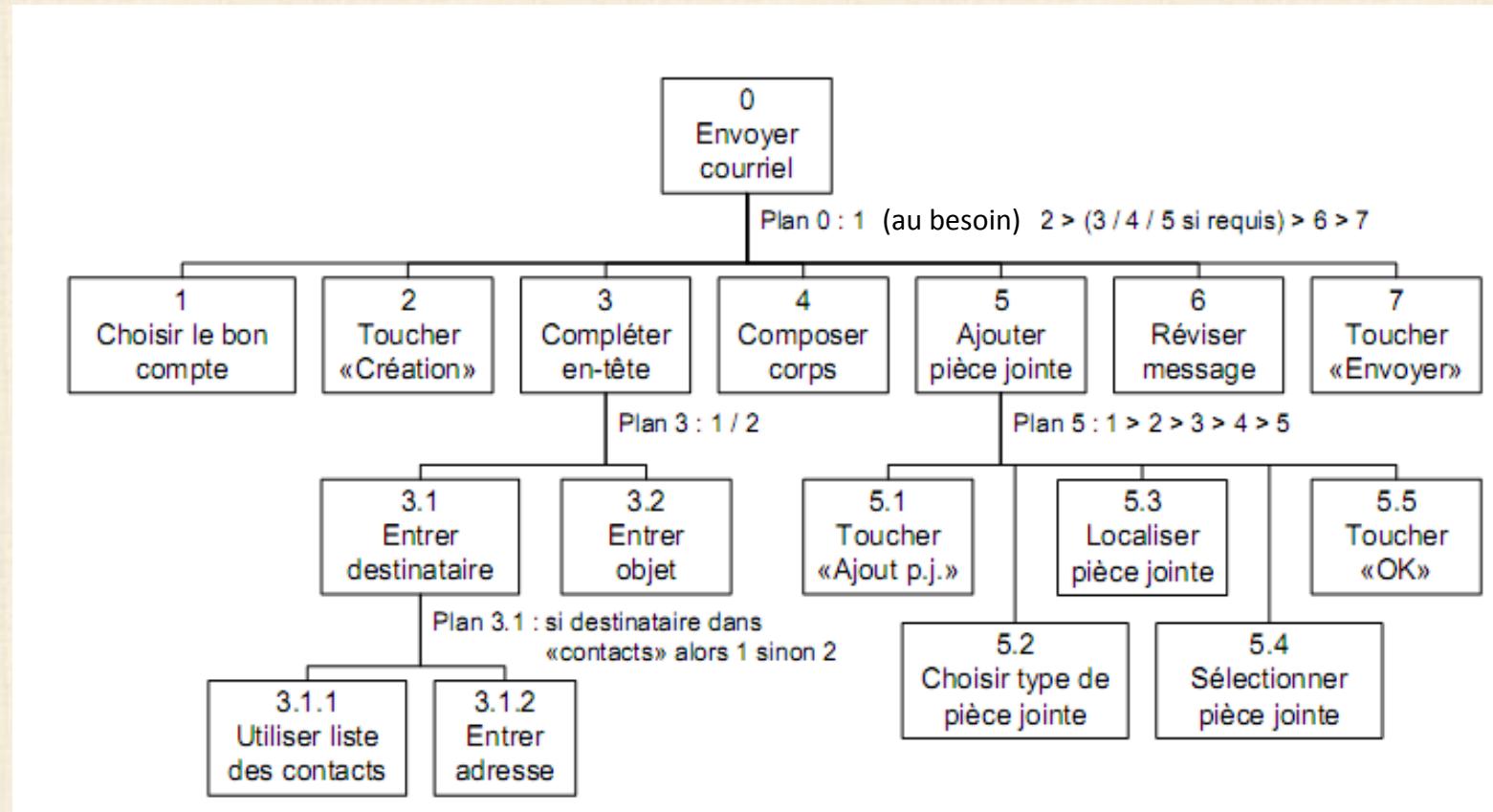
Dans le formalisme HTA les tâches sont numérotées hiérarchiquement comme suit:

- La racine porte le numéro « 0 ».
- Les sous-tâches de la racine portent un nombre entier (« 1 » pour le plus à gauche et successivement en allant vers la droite).

Les sous-tâches de tous les autres nœuds portent un numéro qui résulte de la concaténation du numéro du parent, d'un point et d'un nombre entier (« 1 » pour le plus à gauche et successivement en allant vers la droite).

1) HTA (Cont.)

Exemple : "Envoyer un courriel" en notation HTA :



1) HTA (Cont. Forme textuelle)

Comme tout arbre, une HTA peut être représentée sous la forme d'un tableau (i.e., représentation textuelle). Et par l'ajout de colonnes supplémentaires il est facile de montrer autant de propriétés que souhaité.

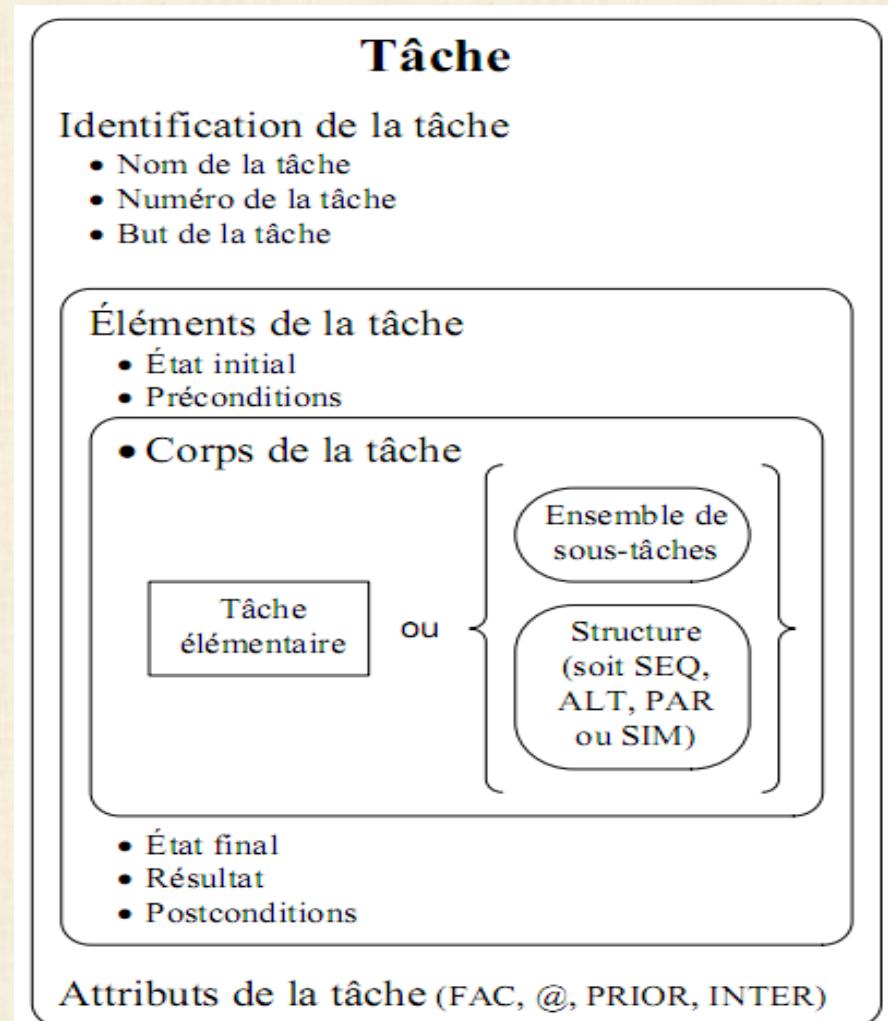
La forme tabulaire (ou textuelle) de l'exemple précédent est la suivante :

Tâches	Type d'entrée	Commentaire
0 Envoyer courriel <i>Plan 0 : 1 (au besoin) > 2 > (3 / 4 / 5 si requis) > 6 > 7</i>		
1 Choisir le bon compte	Sélection	Il faudrait mémoriser ce compte pour la prochaine fois.
2 Toucher «Création»	Clic (bouton)	
3 Compléter en-tête <i>Plan 3 : 1 / 2</i>		
3.1 Entrer destinataire <i>Plan 3.1 : si destinataire dans «contacts» alors 1 sinon 2</i>		
3.1.1 Utiliser liste des contacts	Clic (élément)	
3.1.2 Entrer adresse	Texte	Offrir d'ajouter une nouvelle adresse aux contacts.
3.2 Entrer objet	Texte	
4 Composer corps	Texte	
5 Ajouter pièce jointe <i>Plan 5 : 1 > 2 > 3 > 4 > 5</i>		Att : Utilisation occasionnelle!
5.1 Toucher «Ajout p.j.»	Clic (bouton)	
5.2 Choisir type de pièce jointe	Clic (élément)	
5.3 Localiser pièce jointe	Navigation	Fichier local seulement?
5.4 Sélectionner pièce jointe	Clic (élément)	
5.5 Toucher «OK»	Clic (bouton)	
6 Réviser message		
7 Toucher «Envoyer»	Clic (bouton)	

2) MAD (Méthode Analytique de Description de tâches)

Dans MAD et pour chaque tâche il faut entrer les informations suivantes (indiquées dans la figure à droite):

- Des informations identifiant la tâche, incluant :
 - **un nom et un numéro;**
 - **son but** (en langue naturelle).
- Différentes informations spécifiant les conditions d'exécution de la tâche, incluant :
 - **l'état initial:** l'ensemble des objets permettant l'exécution de la tâche.
 - **les préconditions:** un ensemble de prédicats exprimant des contraintes sur les objets de l'état initial qui doivent être réalisés pour le démarrage de la tâche.
 - le **corps** de la tâche, qui est une tâche élémentaire (indécomposable ou dont le raffinement est jugé suffisant) ou un ensemble de sous-tâches avec une structure temporelle indiquant comment exécuter lesdites sous-tâches. Cette structure est exprimée à l'aide des opérateurs temporelles suivants :

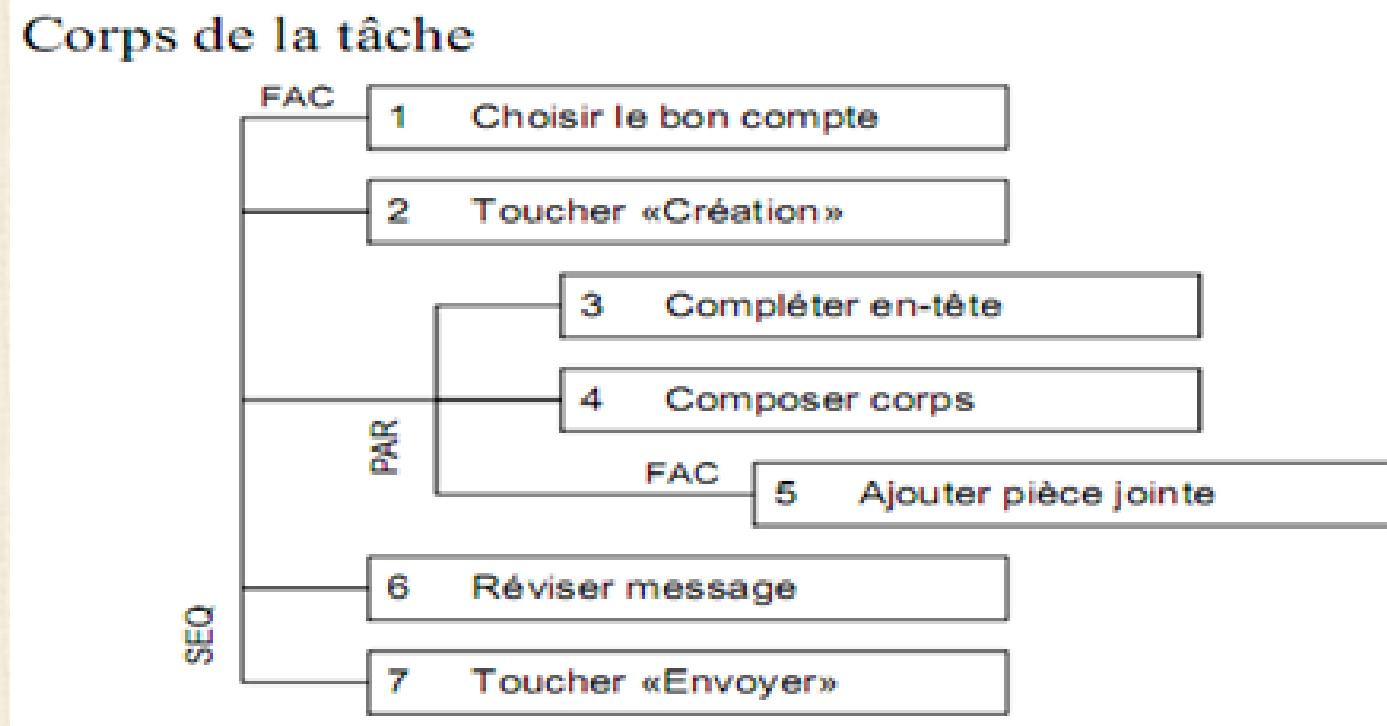


1) MAD (Cont.)

- **SEQ** : si les sous-tâches doivent être exécutées séquentiellement;
- **ALT** : s'il faut ne choisir d'exécuter qu'une seule des sous-tâches;
- **PAR** : si les tâches doivent être exécutées en parallèle par une seule personne;
- **SIM** : si les tâches doivent être exécutées simultanément par plusieurs personnes;
- **ET/OU** : elle est utilisée pour indiquer qu'on peut exécuter autant de tâches que désiré
- **l'état final**: l'ensemble des objets pouvant être créés ou modifiés par la tâche, ainsi que ceux qui doivent exister après son exécution;
- **le résultat**: un sous-ensemble strict de l'état final qui contient tout ce que la tâche a créé ou modifié;
- **les postconditions** : un ensemble de prédictats exprimant des contraintes sur les objets de l'état final qui sont assurément réalisés lorsque la tâche se termine.
- Divers attributs affectant l'exécution de la tâche, incluant :
 - **FAC** : lorsque la tâche est facultative;
 - **@** : lorsque la tâche peut être répétée;
 - **PRIOR et INTER** : ces paramètres définissent l'interruptibilité d'une tâche (en spécifiant une priorité d'exécution qui sera comparée à celle des autres tâches voulant l'interrompre, ainsi que la manière dont l'exécution devra reprendre après l'interruption).

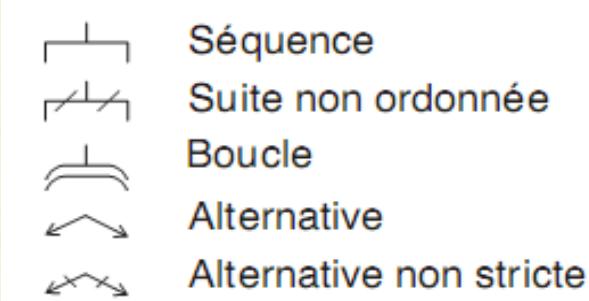
2) MAD (Cont.)

Exemple : « Envoyer un courriel » selon MAD:



2) MAD (Cont.)

Une notation alternative pour MAD:



Exemple: La représentation MAD de l'activité " Gestion des ressources"

