

# Modèles de tâches

Renaud Blanch <blanch@imag.fr>

Université Joseph Fourier, Polytech'Grenoble & UFR IMAG

février 2011

# Plan

- 1 Introduction
- 2 Modèles de tâches
- 3 De l'arbre des tâches à l'interface
- 4 Bibliographie

# Objectifs du cours

Après avoir suivi ce cours, vous saurez :

- **définir** ce qu'est un modèle de tâche ;
- **justifier** l'utilité de cet outil pour l'IHM ; et
- **capturer** l'analyse d'une activité à l'aide d'une notation de modèle de tâche.

# Rappel

## définition

“L'**Interaction Homme-Machine** (IHM) est la discipline qui étudie : la **conception** ; la **mise en œuvre** ; et l'**évaluation** de systèmes interactifs **utiles** et **utilisables** destinés à des humains”.

# Position dans ce cadre

Il faut s'assurer de l'**utilité** dès la **conception**  
(car plus tard, c'est trop tard)

# Position dans ce cadre

Il faut s'assurer de l'**utilité** dès la **conception**  
(car plus tard, c'est trop tard)  
et cela nécessite de **comprendre l'activité** (i.e., le métier)  
instrumenté.

## Position dans ce cadre

Il faut s'assurer de l'**utilité** dès la **conception**  
(car plus tard, c'est trop tard)  
et cela nécessite de **comprendre l'activité** (i.e., le métier)  
instrumenté.  
C'est en cela que consiste l'**analyse de tâches**.

# Méthodes d'acquisition

Il existe diverses **méthodes pour acquérir** cette connaissance de l'activité :

- l'introspection ;
- les questionnaires ;
- les interviews ex-situ ;
- les interviews in-situ ; ou encore
- l'observation en laboratoire.



# Risques lors de l'acquisition

Attention à distinguer lors du recueil ce qui est :  
**normatif** (la manière dont ça devrait se passer) ; de ce qui est  
**effectif** (la manière dont ça se passe en pratique).

# Risques lors de l'acquisition

Attention à distinguer lors du recueil ce qui est :  
**normatif** (la manière dont ça devrait se passer) ; de ce qui est  
**effectif** (la manière dont ça se passe en pratique).

Attention à distinguer les **niveaux d'abstraction** (e.g., «je tape control-b» vs. «je mets en gras» vs. «je tape une lettre» vs. «j'écris à untel»)

# Outils de capture

Pour **transmettre** et **communiquer** cette connaissance, elle doit être capturée.

# Outils de capture

Pour **transmettre** et **communiquer** cette connaissance, elle doit être capturée.

Il existe divers **outils pour capturer** ces informations tels que :

- les scénarios de travail ;
- les *use cases* d'UML ; ou
- les **modèles de tâches**.

# Définitions (1/3)

## définition

Une **tâche** consiste en :

- un **but** (état souhaité) ; et
- une **procédure** pour atteindre ce but.

# Définitions (1/3)

## définition

Une **tâche** consiste en :

- un **but** (état souhaité) ; et
- une **procédure** pour atteindre ce but.

## définition

Une **procédure** est un ensemble de sous-**tâches** liées par :

- des relations de **composition** ; et
- des relations **temporelles**.

# Définitions (2/3)

## définition

Une **tâche élémentaire** est un tâche décomposable en **actions physiques**.

## Définitions (2/3)

### définition

Une **tâche élémentaire** est un tâche décomposable en **actions physiques**.

### définition

Une **action physique** est une opération sur un dispositif d'entrée/sortie qui provoque un changement d'état du dispositif (clic, mouvement, affichage, etc.)



# Définitions (3/3)

## définition

Les **modèles de tâches** sont des structures arborescentes dont :

- les nœuds sont les **buts** ; et
- les sous-arbres sont les **procédures** pour atteindre ces buts.

Les nœuds peuvent être **décorés** par :

- les **concepts du domaine** ;
- les **préconditions** et **postconditions** ;
- la **fréquence** ;
- la **complexité** ;
- la **criticité** ;
- les **contraintes temporelles** ;
- l'**acteur** ;
- ...

# Goals, Operators, Methods, Selection rules (GOMS) [Card et al., 1983]

Il s'agit du premier modèle de tâche comportant un effort de formalisation mais ne gère que des séquences d'actions.

```

GOAL: EDIT-MANUSCRIPT
. GOAL: EDIT-UNIT-TASK ...repeat until no more unit tasks
. . GOAL: ACQUIRE UNIT-TASK ...if task not remembered
. . . GOAL: TURN-PAGE ...if at end of manuscript page
. . . GOAL: GET-FROM-MANUSCRIPT
. . GOAL:EXECUTE-UNIT-TASK ...if a unit task was found
. . . GOAL: MODIFY-TEXT
. . . . [select: GOAL: MOVE-TEXT* ...if text is to be moved
. . . . GOAL: DELETE-PHRASE ...if a phrase is to be deleted
. . . . GOAL: INSERT-WORD] ...if a word is to be inserted
. . . . VERIFY-EDIT
  
```

## *User Action Notation (UAN)* [Hartson et al., 1990]

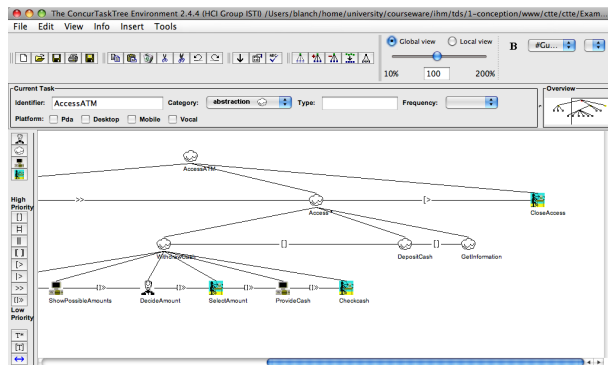
UAN propose un modèle de tâche formalisé par une notation textuelle utilisant les opérateurs de LOTOS.

( )	grouping mechanism
*	iterative closure, task is performed zero or more times
+	task is performed one or more times
{ }	enclosed task is optional (performed zero or one time)
A B	sequence; perform A, then B (same if A and B are on separate, but adjacent, lines)
OR	disjunction, choice of tasks (used to show alternative ways to perform a task)
&	order independence; connected tasks must all be performed, but relative order is immaterial
↔	interleavability; performance of connected tasks can be interleaved in time
	concurrency; connected tasks can be performed simultaneously
;	task interrupt symbol; used to indicate that user may interrupt the current task at this point (the effect of this interrupt is specified as well, otherwise it is undefined, i.e., as though the user never performed the previous actions)



# ConcurTaskTrees (CTT) [Paternò et al., 1997] (1/2)

CTT étend ces opérateurs, et introduit une syntaxe graphique.



## ConcurTaskTrees (CTT) [Paternò et al., 1997] (2/2)

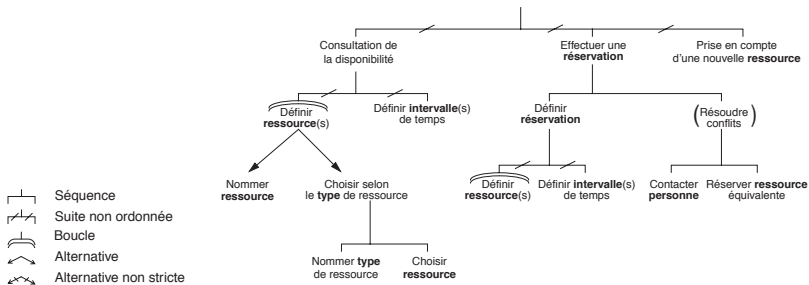
Les opérateurs utilisés par CTT sont :

- $T1 \gg T2$  : la séquence ;
- $T1 [data] \gg T2$  : avec transmission d'information ;
- $T1 [] T2$  : la disjonction ;
- $T^*$  : la fermeture ;
- $T^n$  : la répétition ;
- $[T]$  : l'option ;
- $T1 [ > T2$  : l'interruption définitive ;
- $T1 | > T2$  : l'interruption avec reprise ; et
- $T1 ||| T2$  : le parallélisme.



# Hierarchical Task Analysis (HTA)

HTA est une autre notation graphique, plus simple mais plus limitée que CTT.



# Processus de “concrétisation”

Le processus de “concrétisation” de l'interface à partir du modèle des tâches passe par trois phases, l'élaboration :

- de l'interface **abstraite** ;
- de l'interface **concrète** ; et
- de l'interface **finale**.



# Définitions

## définition

L'**interface abstraite** structure l'interface utilisateur en espaces de travail et spécifie l'enchaînement entre espaces.

# Définitions

## définition

L'**interface abstraite** structure l'interface utilisateur en espaces de travail et spécifie l'enchaînement entre espaces.

## définition

L'**interface concrète** instancie les espaces de travail en fenêtres ou canevas et le contenu des espaces en objets d'interactions.

# Définitions

## définition

L'**interface abstraite** structure l'interface utilisateur en espaces de travail et spécifie l'enchaînement entre espaces.

## définition

L'**interface concrète** instancie les espaces de travail en fenêtres ou canevas et le contenu des espaces en objets d'interactions.

## définition

L'**interface finale** est l'implémentation de l'interface concrète dans le langage de réalisation choisi, pour la plate-forme visée.

# Interface abstraite

- **structurer l'interface** en espaces de dialogue, i.e., partitionner l'arbre des tâches en blocs logiques dont les tailles seront fonction de la plate-forme cible et de la complexité de la sous-tâche ;
- **indiquer les enchaînements** entre espaces ; et
- **lister les concepts** utilisés par chaque espace.




# Interface concrète

- **assigner les espaces de travail** à des fenêtres, canvas (pour l'interaction graphique), etc. ;
- **concrétiser les enchaînements** par le choix d'interacteurs de navigation (onglets, liens, etc.) ; et
- **concrétiser les tâches élémentaires** par le choix d'interacteurs (boutons, champs texte, etc.) et d'interactions.

# Interface finale

- déterminer l'environnement de **programmation** ;
- déterminer l'environnement de **exécution** ; et
- **coder** :)

# Références

-  S. Card, T. M. Moran et A. Newell.  
The Psychology of Human Computer Interaction.  
Lawrence Erlbaum Associates, 1983.
-  H. R. Hartson, A. C. Siochi, and D. Hix.  
The UAN: a user-oriented representation for direct  
manipulation interface designs.  
ACM Trans. Inf. Syst., 8:181–203, July 1990.
-  F. Paternò, C. Mancini, and S. Meniconi.  
ConcurTaskTrees: a diagrammatic notation for specifying  
task models.  
In Proc. INTERACT '97, pages 362–369, 1997.