

Interface Homme Machine

1. Introduction
2. Définitions
3. Critères ergonomiques
4. Étapes du processus de développement et IHM
5. Modèles de tâches
6. Types d'interfaces Homme Machine (Architectures logicielles)
7. Formalismes de validation des interactions sur les IHMs

1. Introduction (Conception- Design)

La technologie de l'interaction homme machine est enrichie récemment des générateurs d'interfaces permettant de créer des maquettes graphiques, tels:

- **Interface Builder**,
- **XFace Maker**
- **et PyQt** (avec son environnement QtDesigner permettant de créer des GUI)

Les interfaces résultantes sont évaluées avec les utilisateurs représentatifs, modifiées de manière itérative jusqu'à ce que les critères d'utilisabilité soient satisfaits.

Utilisabilité* =
(facilité d'utilisation
de l'interface)

Robustesse: elle vise la prévention des erreurs (augmenter les chances de succès):

- Les listes de suggestion.
- Les boîtes de dialogue.
- Désactivation des champs après un certain nombre de tentatives.

+

Souplesse: elle exprime l'ensemble de choix utilisateurs et système:

- Utiliser des icônes au lieu des menus.
- Utiliser des menus contextuels.
- Permettre différentes langues.

1. Introduction (Cont. Interaction)

La conception des systèmes interactifs réalisée en ergonomie et les processus de développement du génie logiciel s'effectuent le plus souvent de manière disjointe:

→ une équipe IHM

→ et une autre pour la conception interne (implémentation)

Alors la collaboration entre ces deux équipes souffre de l'absence de support commun (modèles de la tâche, notations et outils partagés) pour bien achever leurs travaux.

→ Comme solution les spécialistes de l'ergonomie ont inventé de nouveaux modèles et méthodes modélisant l'aspect interactif des systèmes ou applications par exemples:

→ Les formalismes CTT (Concur Task Trees), UAN (User Action Notation) et HTA (Hierarchical Task Analysis) pour construire le modèle de tâches des systèmes interactifs

→ et les méthodes Seeheim, H4, MVC, et PAC, pour modéliser l'architecture logicielle de ces systèmes avec leurs interfaces.

→ Ces solutions IHM sont intégrées aux processus génie logiciel.

1. Introduction (Cont. Technologies -Risques)

→ L'enrichissement des technologies des IHM ont abouti à d'autres soucis ou risques (On doit aussi s'interroger sur son adéquation aux besoins, aux objectifs et aux caractéristiques des utilisateurs):

Le premier risque est :

- D'éliminer les ergonomes des équipes de développement.
 - On dispose des générateurs d'interfaces (Interface Builder, PyQtetc.) qui ne font pas du développeur un spécialiste en ergonomie.

Le second obstacle ou risque est :

- De croire que l'exploitation sauvage des technologies modernes va permettre d'améliorer les interfaces
 - on évoque les systèmes à base des réalités virtuelles, les systèmes multimédias mais aussi les interfaces multi-modales (e.g., Systèmes de sécurité utilisant plusieurs modalités).

2. Définitions

Ergonomie: elle consiste à améliorer l'interface homme machine, en la rendant plus simple et plus logique possible aux yeux et aux caractéristiques de l'utilisateur.

Programme ergonomique: est une application utilisable sans avoir à se référer à l'aide en ligne et diminue au maximum le nombre d'étapes nécessaires à l'utilisateur pour obtenir le résultat recherché.

Une tâche consiste en :

1. **Un but** (état souhaité) ; et
2. **Une procédure** pour atteindre ce but.

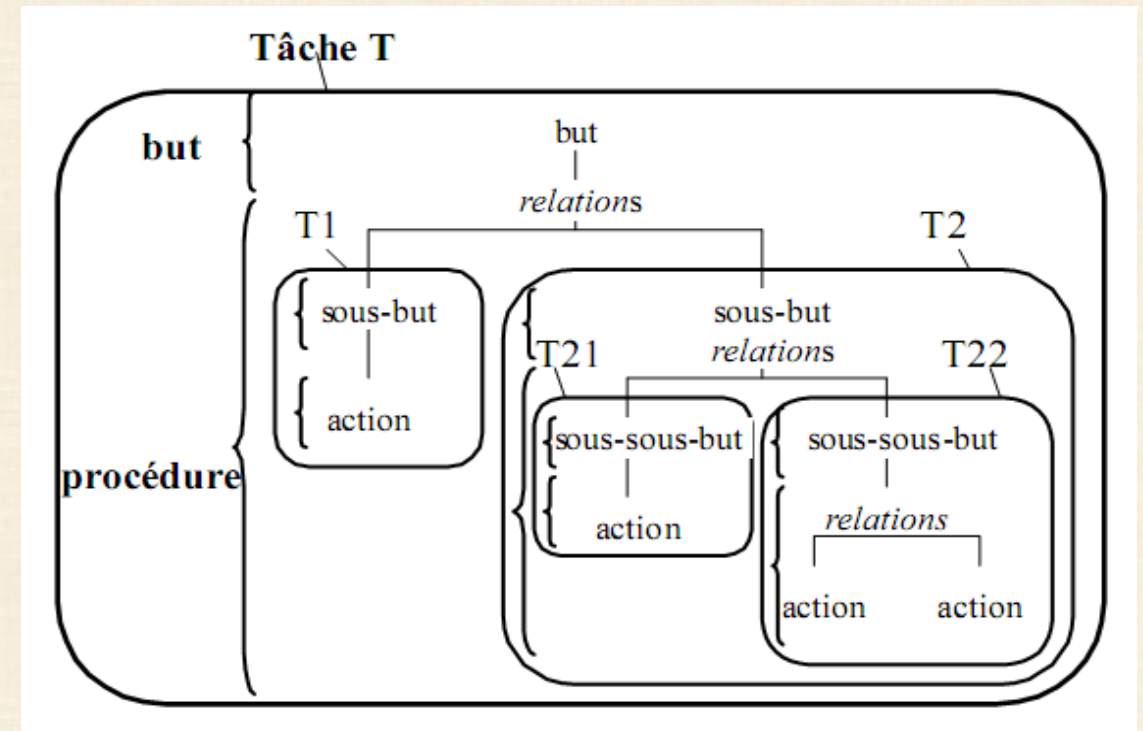
Une procédure est un ensemble de sous-tâches liées par :

- Des relations de composition (T1, T2 filles de T) ; et
- Des relations temporelles (séquentielle, parallèle, entrelacée, alternative stricte/non-strictes....etc).

Une tâche abstraite (i.e., un nœud) est une tâche décomposable en sous tâches ou en tâches élémentaires (atomiques).

Une tâche élémentaire (une feuille de l'arbre) est une tâche qui correspond à une action physique.

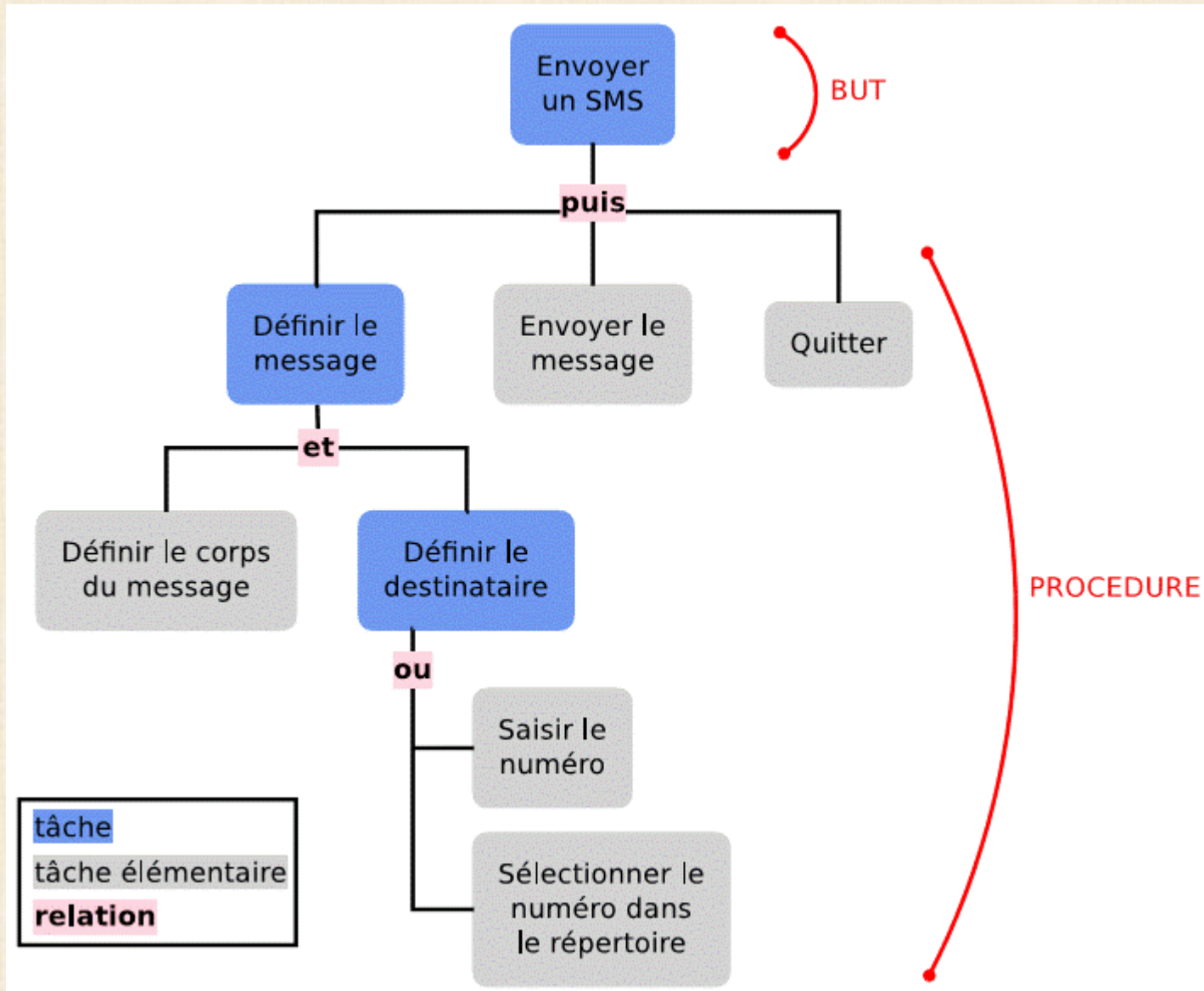
Une action physique est une opération sur un dispositif d'entrée/sortie qui provoque un changement d'état du dispositif (clic, mouvement, affichage, etc.)



Arbre de la tâche T

Exemple d'un arbre de tâches

L'arbre de tâches pour le but 'Envoyer un message'



2. Définitions (Cont.)

Décorations

Elles représentent toute information pertinente permettant d'enrichir l'arbre de tâches

Les nœuds d'un arbre de tâches peuvent être décorés par :

- Les concepts du domaine (objets référencés, i.e. paramètres, variables globales ...)
- Les préconditions et les post-conditions ;
- La fréquence ;
- La complexité ;
- La criticité (niveau de danger, caractère irrévocable) ;
- Les contraintes temporelles (durée maximale) ;
- L'acteur responsable de l'exécution de la tâche (utilisateur et/ou système) ;

2. Définitions (cont.)

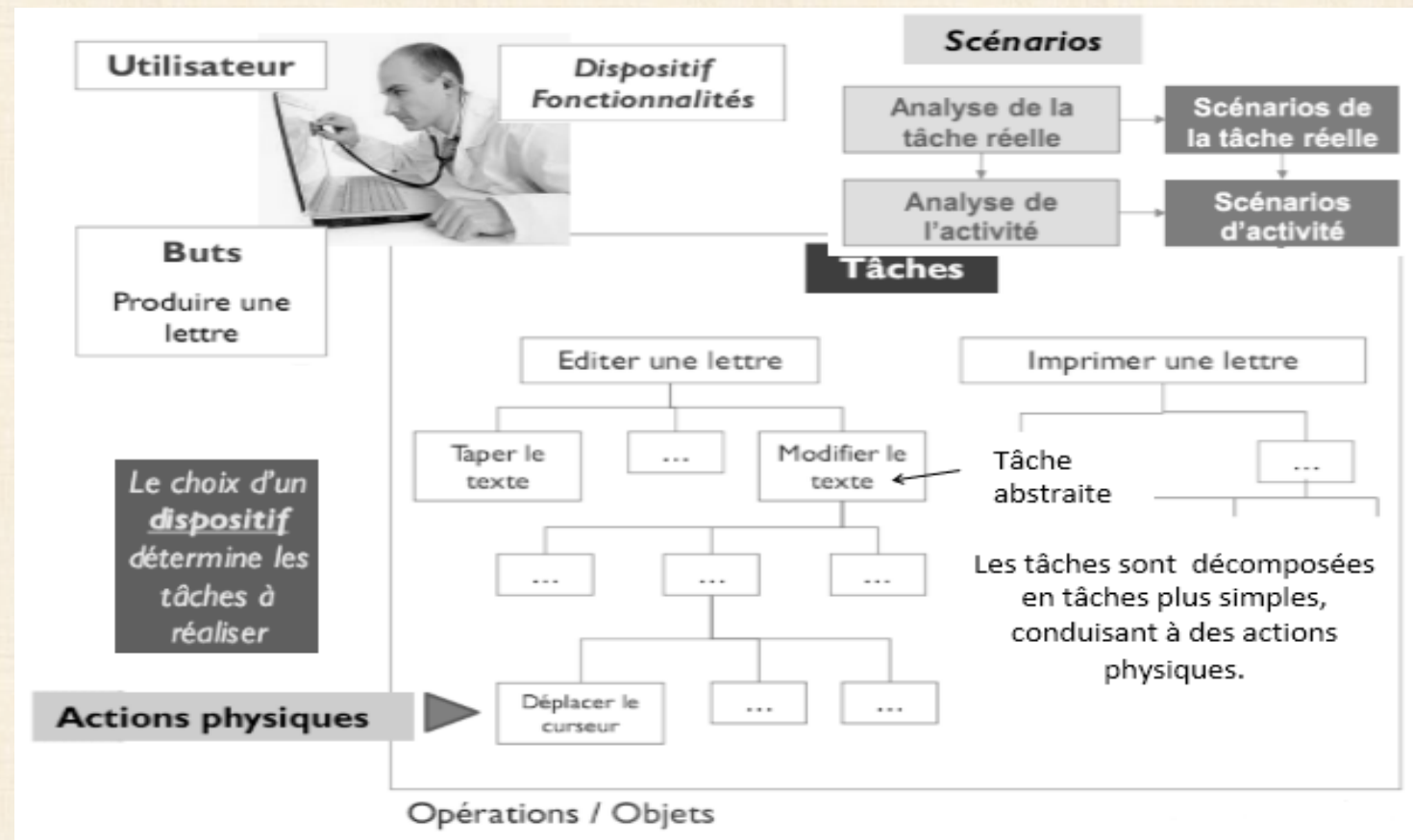
Identification de tâches

Identifier une tâche consiste à :

Analyser
les
besoins
de l'utilisateur

→ Identifier l'objectif recherché par utilisateur lorsqu'il se sert du système.

→ Identifier les informations dont il a besoin.



Pour analyser les besoins de l'utilisateur, on procède généralement en deux étapes :

- 1) Faire des interviews qui permettent d'identifier la tâche prévue (ce que doit être fait);
- 2) Cette analyse est ensuite consolidée par l'observation des utilisateurs afin de comprendre l'activité effectivement réalisée (i.e, faire rédiger des scénarios -**description textuelle des activités de l'utilisateur**- par des utilisateurs représentatifs, ou à défaut soi-même, i.e, l'ergonome).

La connaissance de l'utilisateur permet d'analyser son activité (ce qu'il fait indépendamment du système à l'aide des interviews et des scénarios). Ces résultats sont consignés sous la forme d'un **modèle de tâches**.

2. Définitions (Cont.)

Modèles de tâches

Ce sont des structures arborescentes dont les nœuds sont les buts et les sous-arbres sont les procédures pour atteindre ces buts.

- Ils décrivent comment les activités peuvent être réalisées pour atteindre les objectifs des utilisateurs lors de l'interaction avec l'application considérée.
- Ces modèles permettent d'exprimer les caractéristiques suivantes:
 - la décomposition d'une tâche en sous-tâches (avec éventuellement une typologie de tâche)
 - des relations d'ordonnancement temporel des sous-tâches (séquence, alternative, parallélisme, etc.)
 - les objets utilisés pour accomplir une tâche ou une action,
 - prédire ou expliquer les performances d'un utilisateur dans un environnement donné.

2. Définitions (Cont.)

Exemple de Modèle de tâches : CTT

Concur Task Trees est une notation facile à utiliser, elle permet de représenter les différentes activités de l'utilisateur ainsi que leurs relations temporelles. Elle se focalise sur l'utilisateur en classant ses tâches en quatre types :

- **Abstraite**



- **Utilisateur**



- **Interaction**



- **Application**



CTT utilise un ensemble d'opérateurs temporels pour exprimer les relations temporelles sous la forme d'une grammaire BNF héritée de Lotos* :

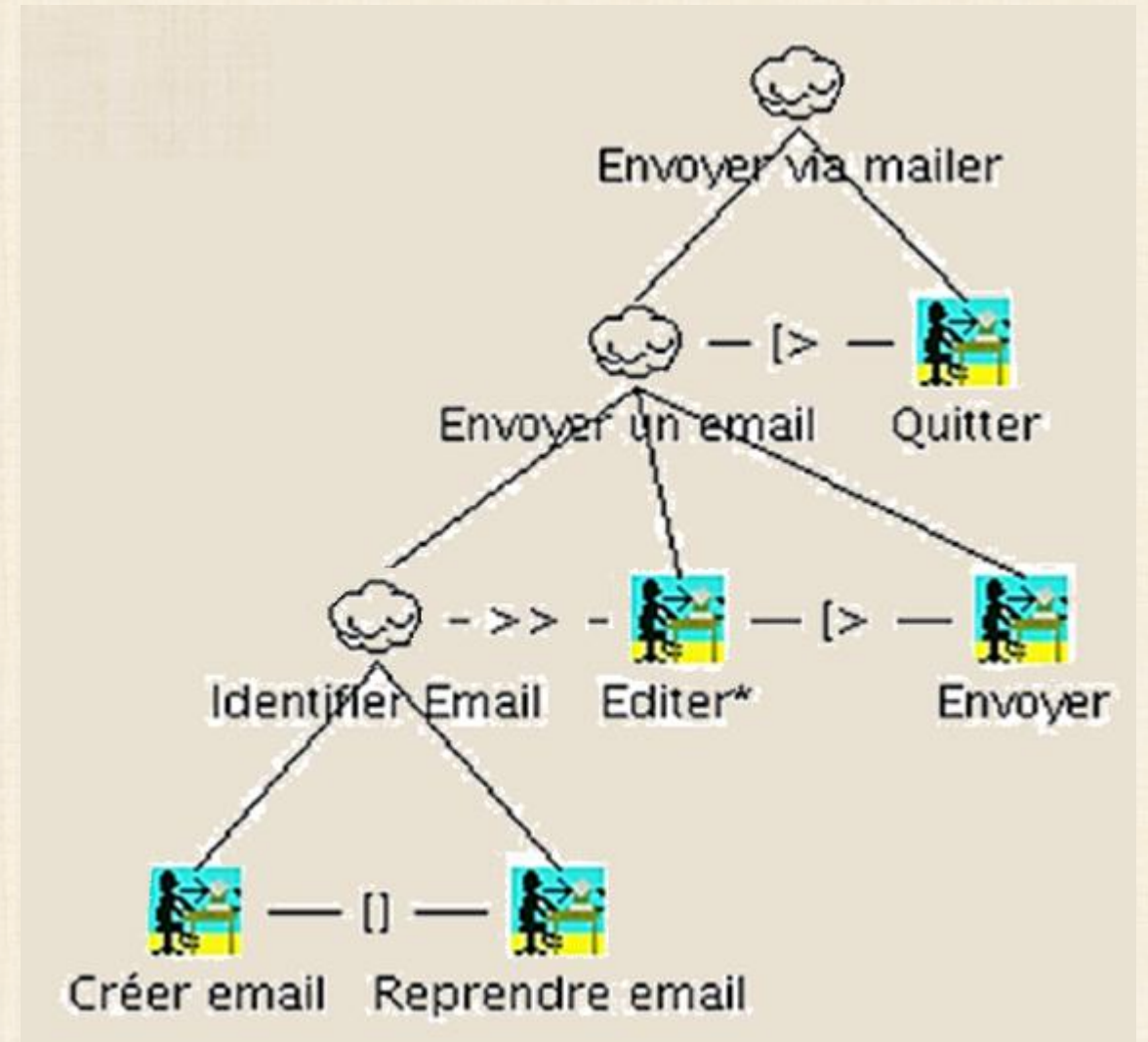
Task ::=	Task >> Task	-- Activation
	Task [] Task	-- Choix
	Task _{At}	-- Tâche atomique
	Task $\models$ Task	-- Ordre indépendant
	[Task]	-- Tâche optionnelle
	Task [> Task	-- Désactivation
	Task* [> Task	-- Désactivation d'une tâche itérative
	Task > Task	-- Interruption
	Task Task	-- Concurrence
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

*Lotos est un langage algébrique pour spécifier l'ordonnancement temporel d'opérations

2. Définitions (Cont.)

CTT utilise une représentation graphique et propose un outil constituant un environnement d'édition, de simulation, et de génération de scénarios de tâches, appelé CTTE* (CTTEEnvironment)

Exemple: Envoyer un message électronique



* <http://giove.isti.cnr.it/lab/research/CTTE/home>

Conception IHM vs génie logiciel:

La conception IHM est basée sur l'aspect ergonomique (visuel et interactif) de l'application par contre la conception génie logiciel se base sur son aspect fonctionnel:

- **Conception IHM**
- S'intéresse aux côtés visuel et interactif de l'application (analyse centrée user):
 - Style de l'utilisateur (Couleurs, forme des fenêtres et boutons ..etc)
 - Tâches de l'utilisateur
 - Fréquence des tâches
 -etc

Comme méthodes de modélisation IHM on a : **CTT**, les formalismes **UAN** et **HTA** (pour modéliser les activités et les tâches) et **Seeheim, Arch, H4, PAC, MVC** (pour modéliser l'architecture logicielle).

- **Conception Génie Logiciel**
- S'intéresse au côté fonctionnel de l'application (analyse centrée implémentation):
 - Les différents modules (fonctions, procédures et classes)
 - Intégration des différents modules
 - Implémentation et tests

Comme méthodes de modélisation génie logiciel on trouve : **UML, E/A**.