

Analyse et modélisation de tâches

1. Introduction

La conception de logiciel interactif (ou conception d'interface homme-machine [IHM], ou conception d'interface) est l'activité qui vise à définir le fonctionnement d'un logiciel destiné à être utilisé par une ou plusieurs personnes. C'est un cas particulier de la conception de produit.

Il ne faut pas confondre la conception de logiciel interactif avec la conception logicielle, phase qui en génie logiciel traditionnel suit la spécification et porte exclusivement sur le découpage d'un programme en composants logiciels. La similitude malheureuse entre les deux dénominations a entre autres des raisons historiques : pendant des décennies, l'interface utilisateur a représenté une partie mineure et peu variée dans les logiciels, et la véritable difficulté était dans la réalisation et la gestion d'un programme informatique.

Il existe de nombreuses méthodes de conception de logiciel interactif. La variété provient du type de produit concerné (paramétrage d'un système d'information, conception d'un jeu vidéo, conception d'un serveur vocal interactif, etc.), des facettes et propriétés du logiciel sur lesquelles se concentre la méthode (interface homme-machine seule, ensemble du produit, utilisabilité, fiabilité, coût de conception, etc.), et de la communauté professionnelle concernée (génie logiciel, ergonomie, designers, spécialistes du Web, spécialistes de l'interface homme-machine). Parmi les méthodes ou classes de méthodes rencontrées dans la littérature, citons : la conception itérative ; la conception centrée sur l'utilisateur ; la conception participative ; la conception contextuelle (contextual design) ; la conception par dérivation de modèles.

2. Analyse des tâches

L'analyse des tâches consiste à relier objectifs, tâches et actions. Il s'agit de comprendre les objectifs des utilisateurs et de comprendre comment ils passent des objectifs aux tâches puis aux actions. Il est également important de savoir ce qui se passe en cas de problème.

Le but de l'analyse des tâches est de produire une liste des tâches et une description des tâches et de leurs interactions.

Le but de l'analyse des tâches est de mettre en évidence et de décrire les tâches à effectuer pour accomplir un travail.

Une analyse très fine des tâches peut également servir à prédire ou à expliquer les performances d'un utilisateur dans un environnement donné.

Dans le domaine des interfaces homme-machine (IHM), cette analyse est une étape essentielle de la conception et de l'évaluation des interfaces. En effet, chaque fonction de l'interface d'une application (chaque interaction) doit concourir à la réalisation d'une tâche ou d'une partie d'une tâche de l'utilisateur. L'analyse des tâches fixe donc un cadre et des objectifs au concepteur d'interfaces.

Une analyse très fine des tâches peut également servir à prédire ou à expliquer les performances d'un utilisateur avec une interface donnée.

2.1. Terminologie

- Un objectif est un état d'un système qu'un agent (humain) souhaite atteindre (écrire une lettre, obtenir un diplôme, etc.)

- Il y a différents moyens (méthodes, outils, etc.) pour atteindre l'objectif.
- La sélection du moyen détermine la tâche à entreprendre.
- Un objectif doit avoir une certaine stabilité dans le temps.
- Une tâche est une activité nécessaire, ou utilisée pour atteindre un objectif en utilisant un moyen donné. Une tâche est en général décomposable en sous-tâches, jusqu'au niveau des actions.
- Une action (ou tâche élémentaire ou *unit task*) est une tâche n'impliquant pas de résolution de problème ou de structure de contrôle (alternatives, répétition, ...). Elle est généralement focalisée sur un ou des objets.

2.2. Modèles de tâches

De nombreux modèles (formalismes) ont été développés (CTT, DIANE+H, GTA, GOMS, MAD, N-MDA (K-MAD en anglais), UAN, ...) dans le but de mettre en évidence et de pouvoir décrire les tâches à effectuer pour accomplir un travail. Ces modèles permettent en général d'exprimer les caractéristiques suivantes :

- la décomposition d'une tâche en sous-tâches (avec éventuellement une typologie de tâche)
- des relations d'ordonnement 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.

Identifier une tâche consiste à

- 1/ identifier l'objectif que cherche à atteindre l'utilisateur lorsqu'il se sert du système et
- 2/ identifier les informations dont il a besoin.

Connaître la tâche permet donc de structurer l'interface selon le point de vue de l'utilisateur.

Pour analyser le besoin des utilisateurs, on procède généralement en deux étapes :

- Des interviews qui permettent d'identifier la tâche prévue (ce qui doit être fait)
- Cette analyse est ensuite consolidée par l'observation des utilisateurs afin de comprendre l'activité effectivement réalisée

Ces résultats sont consignés sous la forme d'un graphe appelé *modèle de tâche*.

2.3. Principaux formalismes et méthodes de modélisation de tâches

2.3.1. GOMS : (Goal, Operator, Method Selection)[Card et al., 83]

Si les modèles de tâches ont essentiellement pour but d'aider à concevoir les interfaces homme-machine, il existe des modèles conçus pour effectuer une analyse cognitive des tâches, dans l'idée de calculer à l'avance ou d'expliquer les performances des utilisateurs. Pour prédire les performances d'un utilisateur, GOMS prend en compte non seulement le temps d'exécution des opérations élémentaires (frappe d'une touche, mouvement de la souris, etc.) mais également l'effort cognitif, et donc le temps nécessaire à la sélection des méthodes (tâches).

```

GOAL: EDIT-MANUSCRIPT
  GOAL: EDIT-UNIT-Task repeat until no more unit tasks
    GOAL: ACQUIRE-UNIT-TASK
      GET-NEXT-PAGE if at end of manuscript
      GET-NEXT-TASK
    GOAL: EXECUTE-UNIT-TASK
      GOAL: LOCATE-LINE
        [select: USE-QS-METHOD
          USE-LF-METHOD]
      GOAL: MODIFY-TEXT
        [select: USE-S-METHOD
          USE-MMETHOD]
      VERIFY-EDIT

```

Limitations de GOMS

- Ne prends pas en compte les erreurs utilisateur
- Ne prends pas en compte les interruptions de tâches
- Ne considère que les tâches séquentielles

2.3.2. UAN - User Action Notation [Hix & Hartson, 92]

2.3.3. MAD : Méthode Analytique de Description de tâches

C'est une méthode orientée Conception d'Interfaces Utilisateur.

Cette définition se retrouve dans de nombreux articles et rapports publiés par Scapin ([Scapin 1988], [Scapin, Pierret-Goldbreich 1989]).

Dans le formalisme MAD, les principaux concepts sont ceux d'objet-tâche, d'action, de structure :

Le concept de tâche est représenté par un objet générique (au sens de frame, catégorie, prototype, etc.) appelé objet-tâche (O.T.). Cet objet comporte les éléments suivants :

- un état-initial (I) : sous ensemble de l'état du monde constitué de la liste des objets, arguments d'entrée de la tâche.
- un état-final (F) : sous ensemble de l'état du monde constitué de la liste des objets, arguments de sortie de la tâche. Il s'agit des objets directement créés ou modifiés suite à l'exécution de la tâche. Certains objets peuvent naturellement apparaître à la fois en entrée et en sortie.
- un but (B) : sous-ensemble de l'état-final (B est inclus dans F), indiquant explicitement le but recherché dans l'exécution de la tâche.
- des préconditions (C.N.) : ensemble des prédicats exprimant des contraintes sur l'état initial et qui doivent nécessairement être satisfaites pour pouvoir déclencher l'exécution de la tâche. On distingue un type particulier de préconditions, les conditions-nécessaires déclenchantes (C.N.D.) qui décrivent des états particuliers qui non seulement doivent être satisfaits pour permettre l'exécution de la tâche mais qui de plus ont un rôle dynamique de déclenchement de la tâche. (Condition nécessaire et suffisante à son déclenchement).
- des postconditions (P.C.) : ensemble des prédicats exprimant des contraintes sur l'état final et qui doivent nécessairement être satisfaites après l'exécution de la tâche. Les postconditions sont des contraintes qui portent sur l'état final.

On distingue deux types d'objet-tâche : l'objet-tâche est donc la racine de deux sous-classes particulières : la classe tâche élémentaire et la classe tâche composée :

- Une tâche élémentaire est une tâche indécomposable dont le niveau opérationnel est caractérisé par un objet méthode de type simple, c'est-à-dire une action. Une tâche élémentaire comporte donc en plus des éléments précédents un descripteur de l'action à exécuter.
- Une tâche composée est une tâche dont le niveau opérationnel est défini par une structure décrivant le corps de la tâche. Celle-ci comporte donc un champ supplémentaire structure.

Le concept de structure est représenté par un objet générique caractérisé par un constructeur et ses arguments constitués d'une liste d'objet-tâche. Les constructeurs décrivent l'agencement des différentes tâches impliquées. Certains constructeurs sont prédéfinis :

- SEQ : tâches séquentielles (par exemple dans le scénario suivant : localiser un fichier, puis déplacer le curseur, puis ouvrir un fichier).
- PAR : tâches parallèles (par exemple dans le scénario suivant : soit insérer des fichiers déjà constitués, puis rédiger un autre texte, soit l'inverse).
- ALT : tâches alternatives (une manière ou une autre de faire; par exemple dans le scénario suivant : effectuer un copie soit par commande clavier, soit par désignation d'options de menu).
- BOUCLE : tâches itératives (par exemple dans le scénario suivant : effectuer des opérations d'insertion de fichier d'un collègue, puis recommencer pour un autre membre de l'équipe).
- FAC : tâches facultatives (pour les opérations non requises, par exemple ajouter un sigle au rapport d'activité de son équipe)."

Sous l'impulsion de l'expansion des modèles objet en informatique, une deuxième définition, esquissée dans [Pierret-Goldbreich et al. 1989], précisée dans [Hammouche 1993], définit les différents éléments d'une tâche MAD sous forme d'objets. Chaque élément devient un objet à attributs. Ainsi une tâche est un objet avec les attributs suivants : nom de la tâche, type, état initial, état final, préconditions, postconditions, structure de sous-tâches. Chaque attribut regroupe en fait des ensembles d'objets. Ainsi l'attribut préconditions est une liste d'objets de type "préconditions". Un objet de type "préconditions" a les attributs suivants : nom, type, prédicats. De même les états initiaux et finaux sont des listes d'objets d'un type appelé "objet" qui contient les attributs nom, type...

MAD a été utilisé sur le terrain dans le cadre de l'analyse de l'activité. La définition a alors été interprétée pour satisfaire les besoins des ergonomes. On peut trouver l'essentiel de ces interprétations dans [Sebillotte 1991]. Elles concernent en particulier les définitions des préconditions et du but.

2.3.4. CTT: ConcurTaskTrees

Le formalisme CTT met l'accent sur les activités de l'utilisateur. Ses spécificités par rapport à d'autres formalismes sont :

1. Une syntaxe graphique
2. Un ensemble de plusieurs opérateurs qualitatifs temporels
3. L'allocation des tâches aux opérateurs
4. Et la présence d'objets et d'attributs pour les tâches

CTT inclut quatre catégories de tâches :

- tâches utilisateurs (activités cognitives ou physiques de l'utilisateur indépendantes de toute interaction sur le système),
- tâches application (tâches effectuées par le système pour rendre compte de l'état du système),
- tâches interaction (tâches réalisées par les interactions de l'utilisateur sur le système) et
- tâches abstraites (tâches raffinées par les tâches précédentes : il s'agit de tâches de haut niveau sans connaître exactement le portée de son action).

L'outil CTTE (<http://giove.isti.cnr.it/CTTE/predownload.html>) a permis la simulation des modèles de tâches exprimés avec la notation. Cette simulation permet de spécifier l'ordre d'exécution des tâches pour un scénario particulier en fonction de relations temporelles exprimées à l'aide des opérateurs CCT.

3. Exercices

3.1. Distributeur de billets

Réalisez le modèle de tâches avec l'environnement CTTE du système de distributeur automatique de billets (DAB).