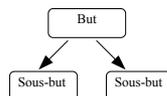


Interaction Humain-Machine

Formalismes
GOMS - Keystroke
CLG

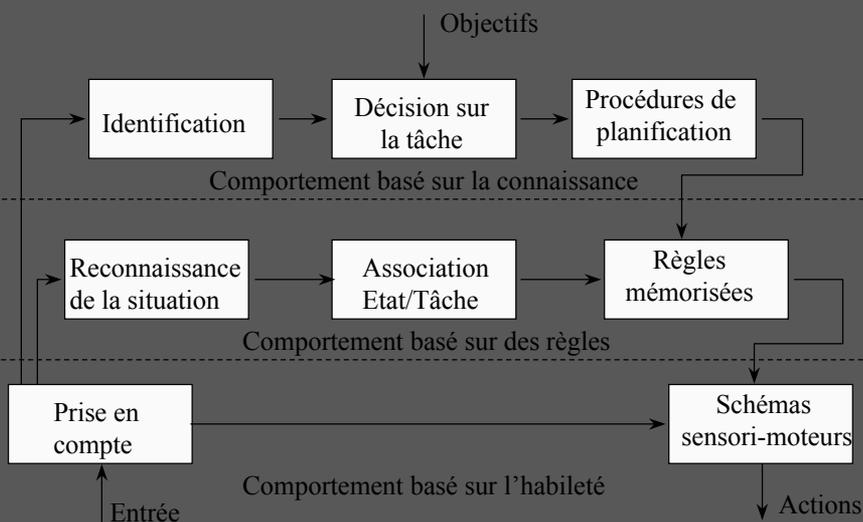
GOMS (Goals, Operators, Methods, and Selection)

- Le formalisme GOMS est un exemple de modèle de description du comportement humain qui prend comme hypothèse le caractère adaptatif mais rationnel du sujet humain.
- Le comportement est représenté par les tâches des utilisateurs, décomposée de façon hiérarchique en sous-tâches, un raffinement en partant de la tâche globale, représenté par des diagrammes de transitions hiérarchiques.
- Ce modèle permet donc la description du comportement humain en différents niveaux d'abstraction (depuis la tâche jusqu'aux actions physiques), et il est basé sur le Modèle du Processeur Humain.



- Le niveau tâche décrit la structure de l'espace de travail en hiérarchie de sous-tâches dépendantes du domaine. Le niveau fonctionnel décrit chaque tâche comme une suite de fonctions, le niveau argument précise chaque fonction par une suite de commandes et le niveau physique spécifie les commandes en termes d'actions physiques.
- Cette méthode peut être utilisée pour la spécification des besoins, pour la conception de l'interface et aussi comme support formel à l'évaluation prédictive de la performance de l'interface dans le cas d'un utilisateur expert. Il permet de prédire le temps d'exécution nécessaire à la réalisation d'une tâche.

Modèle de l'opérateur humain



Niveau Keystroke de GOMS

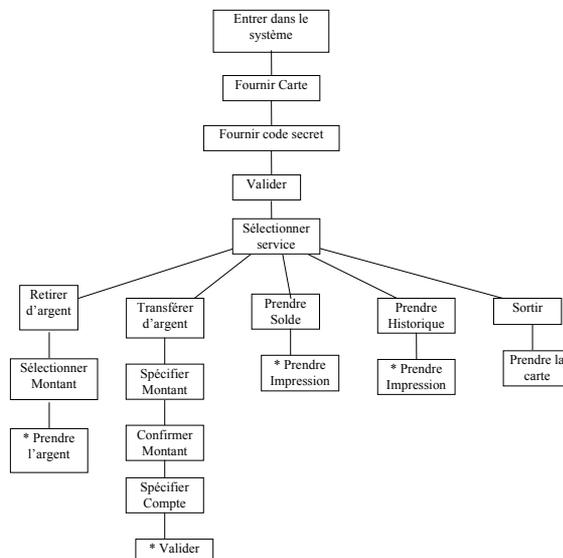
- La méthode Keystroke est un exemple d'utilisation de GOMS comme technique d'évaluation dans la phase de conception. Avec cette méthode il est possible de faire une évaluation de l'utilité et de la performance de l'interface, en regardant le temps d'exécution.
- On spécifie les tâches au niveau physique du modèle GOMS.
- On choisit une méthode de réalisation de la tâche prenant en compte le temps de réponse du système et des caractéristiques des capacités motrices de l'utilisateur par rapport à un style de présentation choisi.

GOMS - Keystroke

- Keystroke permet de prévoir le temps d'exécution (T_e) de la tâche par un utilisateur expert, c'est-à-dire sans erreur ou interruption de la tâche : $T_e = T_k + T_p + T_h + T_d + T_m + T_r$, où
 - T_k (Keystroking) : le temps de frappe au clavier ou de manipulation de la souris
 - T_p (Pointing) : le temps pris pour la désignation de l'objet
 - T_h (Homing) : le temps de rapatriement de la main
 - T_d (Drawing) : le temps pris par l'action de dessiner une trace
 - T_m (Mental activity) : le temps de l'activité mentale de l'utilisateur
 - T_r (Reponse time) : le temps de réponse du système
- En spécifiant le modèle d'accomplissement de la tâche comme une suite de ces opérateurs, Keystroke aide au choix de la conception de l'interface en évaluant sa performance.
- Il faut bien noter les limites de cet outil : l'imprécision de l'opérateur M , la prise des temps moyens et l'absence du temps de réponse du système.

- Dans la même perspective, la Loi de Fitts [ACC 97] peut aussi être utilisée pour prédire la performance des diverses techniques d'interaction dans l'accomplissement des tâches d'acquisition de cible et pour évaluer l'influence de ces paramètres sur l'interaction.

Exemple « Caisse Express »



Application de GOMS

- Initialement mis en place sur des applications de traitement de texte, le modèle GOMS permet, à partir de ces éléments (Goal, Operator, Method, Selection) de représenter l'activité cognitive d'un individu pendant l'accomplissement d'une tâche du système.
- Les buts (GOAL) sont les services que l'utilisateur veut réaliser à travers le système. Ces buts peuvent être décomposés en sous-buts, d'une façon hiérarchique, tout en respectant le modèle humain de partition des problèmes en sus-problèmes lors de leur résolution. Toutefois, dans le modèle GOMS cette hiérarchie n'est pas complètement spécifiée (il y en a des versions qui permettent la représentation de buts en parallèle). Pour montrer son fonctionnement nous nous appuyons sur notre application, pour laquelle nous pouvons identifier les buts suivants :
 - Retirer d'argent
 - Transférer d'argent
 - Prendre le solde
 - Prendre l'historique
 - Sortir du système

GOMS - Opérateurs

- Les opérateurs (OPERATOR) sont des actions réalisées par l'utilisateur pour accomplir un but. Ces **opérateurs** peuvent être **externes**, c'est-à-dire des actions perceptuelles et motrices des utilisateurs sur les objets du système pour changer des informations (actions observées), ou **internes**, c'est-à-dire des opérateurs mentaux des utilisateurs, inférés par les théoriciens et analyseurs.
- Chaque action a une durée propre, qui peut être différente par rapport au niveau d'habileté de l'utilisateur. Les résultats de plusieurs expériences sur différents utilisateurs, dans différents domaines et contextes, ont établi quelques mesures fiables pour ces durées, qui nous prenons en compte.
- Plusieurs lois empiriques du comportement humain, comme la Loi de Hick et la loi de Fitts, ont été donc établies, pour donner des mesures du temps pris par des actions élémentaires réalisées par l'être humain.

Loi de Hicks et Loi de Fitts

- La Loi de Hicks donne une façon de prédire le temps T pris par l'homme pour faire le choix d'un item parmi une liste de n items :

$$T = k \log_2(n+1), \text{ où } k \text{ est une constante}$$

- La Loi de Fitts permet de prédire le temps de pointage T pris par l'homme :

$$T = k \log_2(A / W + 1), \text{ où}$$

A : la distance entre l'objet de pointage et l'icône à être pointé

W : la largeur de l'icône

k : constante (valeur 100 msec)

Opérateurs « Caisse Express »

Description des opérateurs externes GOMS

Opérateur	Description	Temps
H	Ramener la main au clavier	0.40 s
K	Presser une touche	0.20 s
T(n)	Taper une chaîne de caractères	$n \times 0.2$
P	Pointer	1.10 s *
M	Localiser un objet sur l'écran ou vérifier une description	1.35 s
W(t)	Attendre la réponse du système	t_R **
I	Insérer ou retirer une carte	1.50 s

* temps calculé par la Loi de Fitts.

** t_R est le temps pendant lequel le système fait attendre l'utilisateur. Si on considère m comme le temps de traitement d'une commande par le système et t le temps exploité par l'utilisateur pour exécuter un opérateur pendant le traitement de la commande, t_R vaut :

$$t_R = 0 \quad \text{si } m \leq t$$

$$t_R = m - t \quad \text{si } m > t$$

Description des opérateurs internes humains

Description	Temps
Accomplir un but	-
Décider	1.35 s
Rappeler à partir de la mémoire à court terme	1.20 s
Rappeler à partir de la mémoire à long terme	1.35 s
Penser à une valeur	-

- Les méthodes (METHODS) sont des séquences d'opérateurs qui permettent l'accomplissement d'un but. Dans le cas où on pourrait avoir plusieurs méthodes pour le même but, le modèle fournit des règles de sélection de la méthode. Dans notre application, les méthodes sont uniques pour chaque but.

	Opérateur	Durée (s)
Méthode pour le but : Retirer d'argent		
Pas1. Identifier l'écran	M	1.35
Pas2. Décider : si vous n'êtes pas dans le système, accomplir but : Entrer dans le système		
Pas2. Accomplir but : Sélectionner service		
Pas3. Accomplir but : Sélectionner montant		
Pas4. Accomplir but : Prendre d'argent		
Pas5. Retourner avec le but accompli		

Méthodes « Caisse Express » (2)

Méthode pour le but : Entrer dans le système

Pas1. Insérer la carte	I	2.0
Pas2. Attendre vérification de la carte	W(t)	t_R
Pas3. Rappeler de la LTM le code secret		(...)
Pas4. Ramener la main au clavier	H	0.20
Pas5. Tapez code secret	T(4)	1.12
Pas6. Accomplir le but : Valider		
Pas7. Attendre vérification du code	W(t)	t_R
Pas8. Décider, si code incorrecte, aller au Pas3		(...)
Pas9. Retourner avec le but accompli		

Les opérateurs signalés par le symbole (...) sont normalement anticipés dans l'opérateur précédent, leurs durées sont donc comprises dans le temps précédent.

Méthode pour le but : Sélectionner un service

Pas1. Localiser le service sur l'écran	M	1.35
Pas2. Pointer sur l'écran	P	1.10
Pas3. Vérifier que le service a été attendu	M	1.35
Pas4. Retourner avec le but accompli		

Méthodes « Caisse Express » (3)

Méthode pour le but : Sélectionner montant

Pas1. Penser sur le montant désiré	M	1.35
Pas2. Localiser montant sur l'écran		(...)
Pas3. Décider : si montant inexistant, aller au Pas1		(...)
Pas4. Pointer sur le montant	P	1.10
Pas5. Attendre vérification du montant	W(t)	t_R
Pas6. Décider : si montant impossible, aller au Pas1		(...)
Pas7. Retourner avec le but accompli		

Méthode pour le but : Transférer d'argent

Pas1. Identifier écran	M	1.35
Pas2. Décider : si vous n'êtes pas dans le système, accomplir but : Entrer dans le système		
Pas3. Accomplir but : Sélectionner service		
Pas4. Accomplir but : Spécifier montant		
Pas5. Accomplir but : Spécifier compte		
Pas6. Accomplir but : Valider		
Pas7. Attendre le résultat	W(t)	t_R
Pas8. Décider : si " erreur de transaction ", aller au Pas4		(...)
Pas9. Retourner avec le but accompli		

Méthodes « Caisse Express » (4)

Méthode pour le but : Spécifier montant

Pas1. Penser sur le montant à transférer	M	1.35
Pas2. Ramener la main au clavier	H	0.20
Pas3. Tapez le montant	T(n)	n x 0.28
Pas4. Vérifier le montant	M	1.35
Pas5. Décider : si “ montant correcte ”, aller au Pas8		(...)
Pas6. Accomplir le but : Effacer		
Pas7. Retourner au Pas2		
Pas8. Accomplir but : Confirmer		
Pas9. Retourner avec le but accompli		

Méthode pour le but : Spécifier compte

Pas1. Rappeler de la LTM le numéro de compte	M	1.35
Pas2. Ramener la main au clavier	H	0.20
Pas3. Tapez numéro de compte	T (10)	2.80
Pas4. Vérifier la compte	M	1.35
Pas5. Décider : si “ compte correcte ”, aller au Pas8		(...)
Pas6. Accompli le but : Efface		
Pas7. Retourner au Pas2		
Pas8. Retourner avec le but accompli		

Méthodes « Caisse Express » (5)

Méthode pour le but : Valider

Pas1. Lire les valeurs sur l'écran		(...)
Pas2. Localiser le service sur l'écran		(...)
Pas3. Pointer sur “ Valider ”	P	1.10
Pas4. Retourner avec le but accompli		

Méthode pour le but : Effacer

Pas1. Localiser le service “ Effacer ” sur l'écran		(...)
Pas2. Pointer sur “ Effacer ”	P	1.10
Pas3. Attendre le résultat	W(t)	t _R
Pas4. Retourner avec le but accompli		

Méthode pour le but : Confirmer

Pas1. Localiser le service “ OK ” sur l'écran		(...)
Pas2. Pointer sur “ OK ”	P	1.10
Pas3. Attendre le résultat	W(t)	t _R
Pas4. Retourner avec le but accompli		

Méthodes « Caisse Express » (6)

Méthode pour le but : Sortir

Pas1. Localiser le service sur l'écran	M	1.35
Pas2. Pointer sur l'écran	P	1.10
Pas3. Attendre réponse du système	W(t)	t_R
Pas4. Retirer la carte	I	1.5
Pas5. Retourner avec le but accompli		

Dans le but de comparer 2 conceptions différentes par rapport à leurs performances, nous allons supposer maintenant que les services de confirmation, de validation et d'effacement seront fournis par le clavier, et pas par l'écran. Dans cette configuration, nous avons donc les méthodes suivantes (les méthodes qui n'utilisent pas ces services ne sont pas réécrites) :

Méthode pour le but : Valider

Pas1. Lire les valeurs sur l'écran		(...)
Pas2. Ramener la main au clavier	H	0.20
Pas3. Taper sur la touche " Valider "	P	0.28
Pas4. Retourner avec le but accompli		

Méthodes « Caisse Express » (7)

Méthode pour le but : Effacer

Pas1. Ramener la main au clavier	H	0.20
Pas2. Taper sur la touche " Effacer "	P	0.28
Pas3. Attendre le résultat	W(t)	t_R
Pas4. Retourner avec le but accompli		

Méthode pour le but : Confirmer

Pas1. Ramener la main au clavier	H	0.20
Pas2. Taper sur la touche " Valider "	P	0.28
Pas3. Attendre le résultat	W(t)	t_R
Pas4. Retourner avec le but accompli		

La description des méthodes a été réalisée en NGOMSL (Natural GOMS Language [JOH 94]), une parmi les plusieurs variations de GOMS. NGOMSL décrit les méthodes d'une façon similaire qu'un programme, et contient une procédure de développement basée sur la spécification "top-down": décomposition des GOALS jusqu'aux opérateurs élémentaires.

L'intérêt de GOMS

- La conception de l'application en utilisant le modèle GOMS permet de faire une évaluation a priori de l'interface de l'application, en comparant le temps d'accomplissement de chaque tâche.
- Le temps peut être un facteur décisif dans une tâche, et peut donc être déterminant dans le choix d'une méthode de réalisation de la tâche, c'est-à-dire, quels dans le choix du style d'interaction qui permet un accomplissement rapide de la tâche.
- GOMS permet aussi de faire une évaluation des hypothèses envisagées avec l'utilisateur (le temps pris pour la réalisation des opérateurs par les utilisateurs).
- L'évaluation concerne donc principalement la performance de l'interface.

Exemple de comparaison

Tâche à réaliser	Interaction par pointage à l'écran	Interaction par touches clavier
Spécifier et valider le montant 200,00 pour le transfert	$(5,40 + t_R)$ s	$(4,78 + t_R)$ s
Spécifier le montant précédent et le corriger par 2000,00 avant valider	$(9,73 + 2 t_R)$ s	$(8,49 + 2 t_R)$ s
Sortir du système	3,95 s	3,95 s

- Nous pouvons noter que les interactions qui utilisent des touches du clavier sont plus performantes que celles basées sur le pointage à l'écran.
- Par contre, on peut se demander si elles sont plus naturelles et plus claires pour tous les utilisateurs.
- Est-ce qu'il est clair pour tous les utilisateurs qu'il faut confirmer le montant de transfert avant de fournir le numéro de compte ?
- Sauront-ils qu'il est possible de corriger les informations ?
- Pour indiquer clairement les possibilités de services, il est souhaitable d'afficher des messages à l'écran, même si ceci allonge la durée de l'opération par le temps de lecture de ces messages...

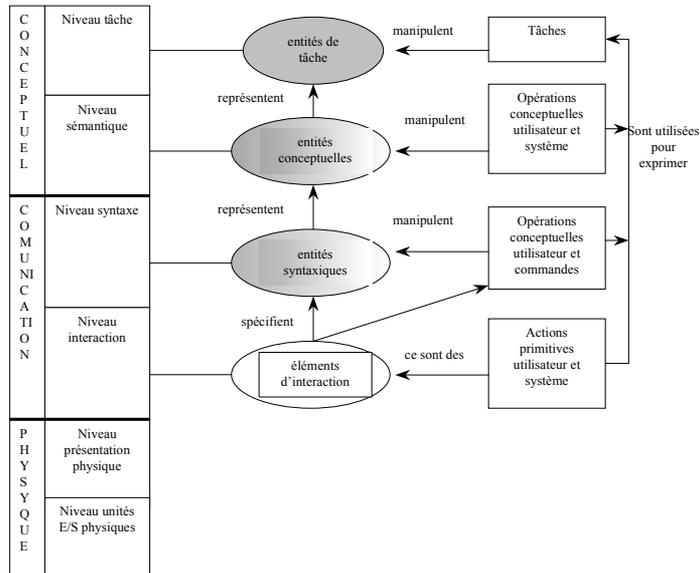
GOMS - le bilan

- Parmi les limites de ce modèle, nous pouvons mentionner le caractère réducteur des opérateurs (nous ne pouvons pas l'utiliser, par exemple, dans une interface de reconnaissance de la parole ou une interface gestuelle), la prise en compte des valeurs moyennes, qui ne correspondent pas toujours à la réalité de l'application (utilisateurs plus ou moins habitué à exécuter les opérateurs), et, surtout, la prise en compte d'un utilisateur " idéal " : celui qui connaît déjà tous les opérateurs, comment les exécuter, les méthodes d'accomplissement de la tâche et ne fait jamais des fautes.
- Ces limites nous empêchent de prendre en compte les traitements d'erreur (correction du montant et du numéro de compte dans le service de transfert) et l'annulation des tâches.
- Les retours d'informations et les réponses du système aux requêtes des utilisateurs ne sont pas non plus prises en compte (nous savons seulement que l'utilisateur doit recevoir une réponse, mais rien sur comment cette réponse est élaborée).
- La prise en compte du comportement de l'utilisateur a donc un caractère très simpliste et même incorrect car nous ne pouvons pas prendre en compte le tâtonnement des utilisateurs novices, cas courant dans notre application.

CLG (Command Language Grammar)

- CLG, comme le modèle GOMS, permet de structurer l'analyse d'une interface par affinements successifs, du niveau tâche jusqu'au niveau des actions physiques. Il permet en plus d'exprimer les relations temporelles entre tâches et les spécifications de valeurs par défaut, ce que permet une évaluation prédictive de la facilité d'utilisation de l'interface et de localiser les décisions d'évaluation qui seront critiques pour les utilisateurs.
- Comme d'autres modèles de conception, CLG ne permet pas l'expression du parallélisme entre les tâches et des interruptions. Ce formalisme est donc pertinent dans les domaines comme ceux des activités de contrôle, où les interruptions sont rares.
- Ce modèle a été conçu pour permettre la représentation complète d'un système informatique, de façon à permettre une représentation de chaque étape de la conception du système et aussi de décrire la représentation mentale que l'utilisateur a (ou qu'on suppose qu'il a) du système, tout dépend du point de vue qui on prend en compte (psychologie de l'utilisateur ou du concepteur). Le modèle de Moran structure le système informatique en trois composantes, comme nous montre la figure ci-après :

Le modèle de Moran d'un système



25

CLG

- Le modèle de Moran représente un système informatique à différents niveaux d'abstraction, où chaque niveau est une représentation complète du système, un raffinement de la vue immédiatement supérieure de la hiérarchie.
 - les niveaux tâche et sémantique pour la composante conceptuelle, donc pour les premières étapes de conception de l'application ;
 - les niveaux syntaxe et interaction pour la conception logique de l'interface ;
 - les niveaux présentation et unités d'E/S pour la composante physique.
- Le passage d'un niveau à l'autre permet de raffiner la conception jusqu'au niveau de détail voulu. Les éléments qui permettent de faire ce raffinement sont les entités (objets manipulés), les opérations élémentaires de manipulation des entités, les procédures (composition d'opérations élémentaires et les méthodes pour accomplir les tâches).

26

CLG « Caisse Express » (1)

■ *Le niveau tâche*

- Ce niveau représente l'utilité du système, les besoins et les exigences de l'utilisateur. Il décrit donc les services que le système doit fournir à l'utilisateur. Pour cela, il faut spécifier les éléments de ce niveau.
- Dans l'application *Caisse Express* nous avons **les entités** :
 - ➔ carte bancaire,
 - ➔ code secret,
 - ➔ montant de retrait,
 - ➔ montant de transfert,
 - ➔ numéro de compte destinataire,
 - ➔ solde,
 - ➔ historique.

CLG « Caisse Express » (2)

Carte = (une entité NOM = " carte bancaire "

(* carte bancaire fournie par l'utilisateur au système. La carte peut être locale ou externe à la banque, des informations qu'elle contient en dépendent.))

Code = (une entité NOM = " code secret "

(* le code secret du client correspond à la carte fournie. Ce code est numérique.))

Montant = (une entité NOM = " montant d'argent "

(* somme d'argent que l'utilisateur veut prendre pour lui ou transférer à partir de son compte vers un autre compte local))

Solde = (une entité NOM = " solde "

(* somme d'argent présente sur la compte de l'utilisateur ou sur sa carte externe.))

Historique = (une entité NOM = " historique "

(* ensemble d'activités bancaires réalisées par l'utilisateur pendant les 30 derniers jours.))

Compte = (une entité NOM = " numéro de compte du destinataire "

(* numéro de compte donné par l'utilisateur comme destination d'un transfert.))

CLG « Caisse Express » (3)

- **Les tâches** : retirer de l'argent, transférer de l'argent, connaître le solde, connaître l'historique, sortir du système, annuler transaction.
- **Les méthodes et procédures** des tâches : les tâches sont décomposées en sous-tâches, jusqu'aux tâches élémentaires et l'espace de travail est structuré par rapport au modèle conceptuel de l'utilisateur.

Retirer-argent = (une tâche (* consiste à faire sortir l'argent voulu)

RESULTAT = montant

PARAMETRE (Carte, Code)

FAIRE EN SEQUENCE :

(SELECTIONER-SERVICE-RETIRER

SPECIFIER-MONTANT

VALIDER)

CLG « Caisse Express » (4)

Transférer-argent (une tâche (* consiste a transférer l'argent d'un compte de l'utilisateur vers un autre compte local)

RESULTAT = Montant, Compte

PARAMETRE (Carte, Code)

FAIRE EN SEQUENCE :

(SELECTIONER-SERVICE-TRANSFERT

SPECIFIER-MONTANT-TRANSFERT

SPECIFIER-COMPTTE

VALIDER)

- Nous pouvons noter que le niveau tâche n'entre pas dans le système. Il décrit l'utilité du système aperçu par l'utilisateur : les tâches sont celles réalisées par l'utilisateur, et décomposées d'une façon naturelle pour lui.

CLG « Caisse Express » (5)

➔ Niveau Sémantique

- Ce niveau est une définition conceptuelle des entités tâches, opérations et méthodes définies au niveau précédant mise en jeu par le système. Les méthodes sémantiques et les entités conceptuelles concrétisent le lien entre le fonctionnement du système et le domaine d'application, et sont encore décrites telles que l'utilisateur les perçoit (et non tel que le système les modélise).

Caisse-Express = (UN SYSTEME
ENTITES : Montant Solde Historique Carte ...)

OPERATIONS = (ENSEMBLE : AFFICHER CHERCHER IMPRIMER LIRE
VERIFIER SPECIFIER)

Montant-RETRAIT = (UNE ENTITE
REPRESENTE(UN Montant-RETRAIT)
NOM = " somme d'argent à retirer "
VAL = (UN ENTIER))

CLG « Caisse Express » (6)

Montant-TRANSFERT = ((UNE ENTITE
REPRESENTE (UN Montant)
NOM = " somme d'argent à transférer "
VAL = (UN REEL))

Historique = (UNE ENTITE
REPRESENTE (UN Historique)
DATE-DEBUT = (UNE DATE)
DATE-FIN = (UNE DATE)
DESCRIPTION = (UNE Transaction))

Transaction = (UNE ENTITE
COMPOSANT-DE (UN Historique)
(ENSEMBLE : UNE CHAINE-ALPHABETIQUE ET UNE DATE))

CLG « Caisse Express » (7)

- Les **opérations conceptuelles** sont des actions élémentaires applicables par le système ou par l'utilisateur aux entités conceptuelles. Parmi elles, nous pouvons mentionner :

AFFICHER (UNE OPERATION-SYSTEME

(OBJET = (UNPARAMETRE VALEUR = (UNE ENTITE) DANS UN PARAMETRE VALEUR = (UN EMPLACEMENT DE ECRAN)))

LIRE (UNE OPERATION UTILISATEUR

(OBJET = (UN PARAMETRE VALEUR = (UNE ENTITE) DANS UN PARAMETRE VALEUR = (UN EMPLACEMENT DE ECRAN)))

CLG « Caisse Express » (8)

- Les **méthodes et procédures sémantiques** se composent de la combinaison et de la composition des procédures des tâches élémentaires. Ils servent à décrire un moyen pour accomplir une tâche.

Sem1 = (UNE METHODE-SEM POUR SPECIFIER-SERVICE-TRANSFERT

RESULTAT = (UN SERVICE)

FAIRE EN SEQ :

(AFFICHER (SERVICES))

(CHERCHER (SERVICE-TRANSFERT) APPARTENANT-A LISTE DE SERVICES-POSSIBLES)

(SI (SERVICE-TRANSFERT) EST-PRESENT SUR LISTE DE SERVICES POSSIBLES ALORS

((SPECIFIER SERVICE)

(AFFICHER(LE RESULTAT DE (SPECIFIER SERVICE))))))

Sem2 = (UNE METHODE-SEM POUR SPECIFIER-MONTANT-TRANSFERT

Montant = (UN PARAMETRE VALEUR = (UN Montant-TRANSFERT)

RESULTAT = (Montant-TRANSFERT = (LE Montant)

FAIRE EN SEQ :

(OPTIONNELLEMENT(CHERCHER (Sasie-Montant)))

(TAPER (Montant))

(AFFICHER(Montant))

(LIRE(Montant))

(CONFIRMER (Montant)))

CLG « Caisse Express » (9)

- Le **niveau sémantique**, comme le niveau tâche, utilise un langage naturel pour décrire les tâches du domaine, en passant d'un niveau orienté utilisateur (niveau précédent) au niveau orienté système. Par contre, cette description conserve l'activité de l'utilisateur, car il y est inclus non seulement la participation du système que l'utilisateur perçoit, mais aussi la participation propre de l'utilisateur.
- Les descriptions conceptuelles doivent être complétées par l'aspect communication système-utilisateur, c'est à dire en précisant la présentation du système, la façon que l'utilisateur doit agir pour obtenir les services du système et la façon que le système réagit aux requêtes. Cette description de présentation est réalisée par les niveaux suivants.

CLG « Caisse Express » (10)

➔ **Niveau Syntaxique**

- Le niveau syntaxique définit la structure du langage de commande à travers des éléments suivants : les commandes (requêtes de l'utilisateur au système), les arguments (entités conceptuelles référencées dans une commande), les contextes, les variables d'état, les canaux de sortie et les méthodes syntaxiques.
- Les arguments sont définis par sa forme, c'est à dire un type de valeur, une valeur, une valeur par défaut, une notation employée par l'utilisateur et une procédure d'interprétation par le système. Nous avons ci-dessous un exemple d'un argument :

```
ID-Montant-TRANSFERT (UN DESCRIPTEUR
FORME = (UN LABEL)
VALEUR = (UN Montant-TRANSFERT)
PROCEDURE D'INTERPRETATION
(SI (FORME(PAS(MEME-FORME QUE(Montant-TRANSFERT))) ALORS (*notifier Montant
invalide))))
```

CLG « Caisse Express » (11)

- Les contextes
- Le contexte doit être vu par le concepteur comme une extension de la mémoire à court terme de l'utilisateur, car il représente l'ensemble de variables psychologiques que l'utilisateur doit mémoriser pour un travail spécifique (variables d'état du système, les commandes permises à partir d'un contexte, les commandes qui entraînent un contexte, etc.). Dans notre application, nous pouvons identifier 5 contextes : le contexte initial, le contexte de composition de code, le contexte principal, le contexte de retrait et le contexte de transfert.

Contexte-principal = (UN CONTEXTE
 VARIABLES D'ETAT = ENSEMBLE : Date-Jour, Montant-Limite, Montant-Minimal Solde-Actuel, Services-Possibles
 ZONES D'AFFICHAGE = (ECRAN-SERVICES)
 COMMANDES-D'ENTREE = (VALIDER-CODE)
 COMMANDES-DE-SORTIE = (RETRAIT SOLDE HISTORIQUE TRANSFERER SORTIR)
 DESCRIPTEURS = (ID-Carte, ID-Code))

CLG « Caisse Express » (12)

Contexte-Retrait = (UN CONTEXTE
 VARIABLES D'ETAT = ENSEMBLE : Date-Jour, Montant-Limite, Montant-Minimal Solde-Actuel, Montant-Retrait
 ZONES D'AFFICHAGE = (ZONE-Message, ZONE-Montants-Possibles)
 COMMANDES-D'ENTREE = (Retirer-argent)
 COMMANDES-DE-SORTIE = (ANNULER)
 DESCRIPTEURS = (ID-Montant-RETRAIT))

Contexte-Transfert = (UN CONTEXTE
 VARIABLES D'ETAT = ENSEMBLE : Date-Jour Solde-Actuel Montant-TRANSFERT, Compte-Destin, Montant-Limite)
 ZONES D'AFFICHAGE = (ECRAN-Transfert = ZONE-Message, ZONE-Sasie-Montant, ZONE-Saisie-Compte, ZONE-confirmation)
 COMMANDES-D'ENTREE = (Transférer-argent)
 COMMANDES-DE-SORTIE = (ANNULER)
 DESCRIPTEURS = (ID-Montant-TRANSFERT, ID-COMPTE))

- Les canaux de sortie sont les parties d'information que le système destine à l'utilisateur dans le cadre d'un contexte. Dans notre application, ces canaux sont les zones d'écran, les différents écrans et l'impression (pour le solde et l'historique).

CLG « Caisse Express » (13)

- Maintenant, nous pouvons identifier dans chaque contexte les commandes proposées à l'utilisateur.

Commande-principale = (UNE COMMANDE CONTEXTE = Contexte-principal)

Commande-confirmation = (UNE COMMANDE CONTEXTES = Contexte-Transfert Contexte-Retrait)

SORTIR (UNE Commande-principale NOM = " Sortir du système "

EFFET (EN SEQ :
(AFFICHER (message) DANS ZONE-Message)
(JETER (Carte))
(ENTRER DANS CONTEXTE-Initial))

SELECTIONNER-SOLDE (UNE Commande-principale NOM = " Solde "

EFFET (EN SEQ :
(AFFICHER (message) DANS ZONE-Message)
(IMPRIMER (Solde))
(AFFICHER (ECRAN-PRINCIPALE))

SPECIFIER-MONTANT-TRANSFERT (UNE Commande-Transfert

EFFET (EN SEQ :
(SI (PAS(DANS Contexte-Transfert)) ALORS (ENTRER DANS LE Contexte-Transfert)
(AFFICHER (ECRAN-Transfert))
(Objet1 = UN ARGUMENT FORME = (UNE ID-Montant-TRANSFERT)
(AFFICHER Objet1 DANS ZONE-SAISIE-Montant)

(CONFIRMER(Objet1)

(LIER Montant-TRANSFERT Objet1))

CLG « Caisse Express » (14)

VALIDER-TRANSFERT (UNE Commande-Transfert NOM = " Valider transfert "

EFFET
(SI (EST-UN (FORME Objet1) LABEL-Montant) ET
(EST-UN (FORME Objet2) LABEL-Compte) ET
(ALORS ((CONSULTER (Montant-limite))
(CONSULTER(Compte))
(SI (Compte ET Montant) = " OK " ALORS ((ENTRER Contexte-principal)
(AFFICHER (ECRAN-principale)))
(SINON (AFFICHER (Message) DANS ZONE-Message))))

EFFET-DE-BORD =
((LIER Montant-Transfert Objet1)
(LIER Compte Objet2))

ANULER (UNE Commande-transfert, UNE Commande-retrait NOM = " Alunner "

EFFET (EN SEQ :
(ENTRER DANS LE CONTEXTE-Principal)
(AFFICHER (Ecran-principale))

CLG « Caisse Express » (15)

- Les **méthodes et procédures syntaxiques** sont constituées de commandes et d'opérations conceptuelles :

Smeth1 = (UNE METHODE-SYN POUR SPECIFIER-MONTANT-TRANSFERT

Montant = (UN PARAMETRE VALEUR = (UN Montant-TRANSFERT)

FAIRE EN SEQ :

(OPTIONNELLEMENT (CHERCHER ZONE-Message)

(LIRE (Message) DANS ZONE-Message))

(CHERCHER (Sasie-Montant) DANS ECRAN-Tansfère)

(TAPER (Montant) DANS ZONE-Saisie-Montant)

(AFFICHER(Montant) DANS ZONE-Saisie-Montant)

(LIRE (Montant) DANS ZONE-Saisie-Montant)

(CONFIRMER (Montant))

CLG « Caisse Express » (16)

- Ce niveau permet d'identifier :
 - Quelles opérations conceptuelles sont permises à un instant donné, à travers l'association entre commandes et contextes.
 - Quelles entités conceptuelles doivent être données à une commande spécifiée, selon la forme de désignation que l'utilisateur perçoit, à travers l'association des commandes et leurs arguments
 - Quelles sont les informations fournies à l'utilisateur et quels supports de présentation sont disponibles dans un contexte spécifique, à travers de l'association canaux et contextes
 - Quelles sont les informations utiles aux diverses étapes du dialogue qui sont mémorisées par le système, de façon à libérer l'utilisateur de cette charge. Cette donnée est présente dans l'association entre les contextes et les variables d'état.
- Avec les notions de contexte et de commande, le niveau syntaxique définit la structure générale de l'interaction entre l'utilisateur et le système, mais sans spécifier les actions physiques élémentaires nécessaires à la spécification des commandes.

CLG « Caisse Express » (17)

- Le **niveau Interaction** décrit en termes d'actions physiques comment l'utilisateur et le système interagissent. Dans notre application, l'ordre peut être résumé de la façon suivante :

CONTEXTE INITIAL

INTRODUIRE-Carte

CONTEXTE CODE

COMPOSER-Code

VALIDER-Code

CONTEXTE PRINCIPAL

RETRAIT

CONTEXTE RETRAIT

VALIDER-Montant

EFFACER

ANNULER

SOLDE

HISTORIQUE

TRANSFERT

CONTEXTE TRANSFERT

VALIDER-TRANSFERT

EFFACER

ANNULER

SORTIR

EFFACER

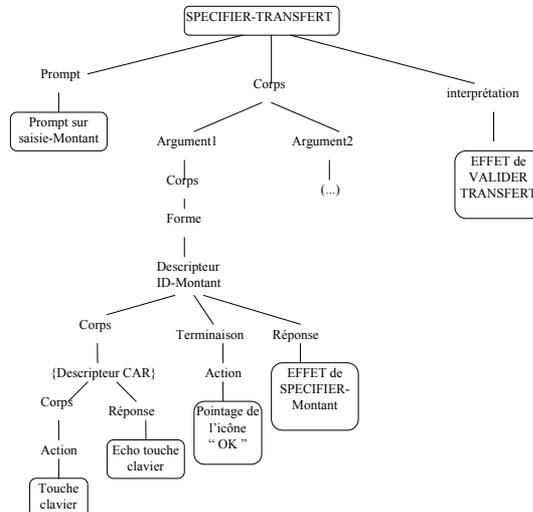
ANNULER

CLG « Caisse Express » (18)

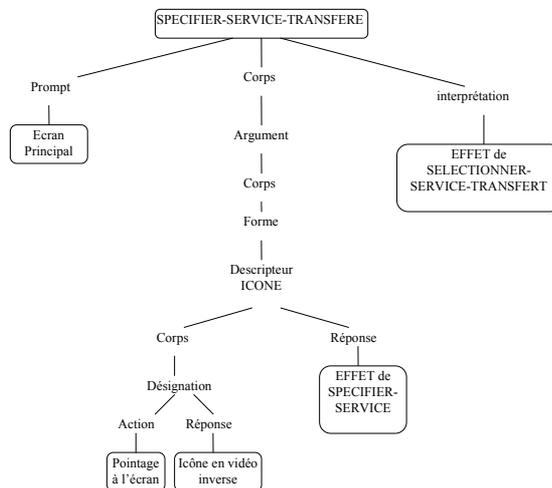
- Les actions sont décrites à travers les éléments d'interaction, qui peuvent être terminaux ou non terminaux.
- Les éléments terminaux, actions produites par le système ou par l'utilisateur sont :
 - ➔ Actions du système : un Prompt (P) et une Réponse ou un Accusé de réception de la spécification faite par l'utilisateur (R).
 - ➔ Actions de l'utilisateur : actions élémentaires (A) comme frappe de touches du clavier, pointage du doigt sur l'écran, déplacement de la souris.
- Un élément non terminal est soit un corps de spécification (C), soit la terminaison de la spécification (T), soit une procédure d'interprétation d'une spécification (I).
 - ➔ C : le corps d'une commande est sa Désignation parmi toutes les commandes possibles dans le contexte, et le corps d'un argument comprend sa Désignation et sa Forme.
 - ➔ T : signale la fin de la spécification d'un élément (par exemple, un retour chariot).
 - ➔ I : désigne la procédure d'interprétation associée à un élément et précise quand elle doit être exécutée.

CLG « Caisse Express » (19)

- Les schémas ci-dessous montrent les spécifications de 2 commandes de la « Caisse Express » :



CLG « Caisse Express » (20)



CLG « Caisse Express » (21)

■ Les règles d'interaction

En CLG, les règles servent :

- ➔ à préciser l'ordre de spécification des éléments syntaxiques,
- ➔ à indiquer quand le système doit les interpréter
- ➔ à définir les éléments d'interaction sous la forme d'actions élémentaires.

■ Parmi les règles de notre application, nous pouvons mentionner :

R-COMMANDE1 (REGLE POUR (QUAND SPECIFIER (LA COMMANDE ANNULER)
(N-IMPORTE-QUAND DANS TOUT CONTEXTE)))

R-COMMANDE2 (REGLE POUR (QUAND INTERPRETER-SPECIFICATION DE (LA COMMANDE
ANNULER)
(TOUT SUITE APRES DESIGNATION DE LA COMMANDE)))

R-COMMANDE3 (REGLE POUR (ACTION DE DESIGNATION DE (UNE COMMANDE
PRINCIPALE)

(DOIGT : placer-doigt-sur (ICONE DE LA COMMANDE)
rester pendant au moins (0.20sec*)))

CLG « Caisse Express » (22)

- * La pointage doit avoir une durée qui permet au système de l'identifier. Malgré la dépendance de cette durée par rapport aux caractéristiques des équipements du système, nous avons pris en compte une durée similaire à celle d'un click rapide de la souris.
- Les méthodes et procédures d'interaction détaillent celui du niveau syntaxique en remplaçant les commandes par les actions physiques introduites par les règles d'interaction. Pour exemple, nous utilisons le méthode Smeth1 du niveau syntaxique :

Imethod1 = ((UNE METHODE-D'INTERACTION POUR SPECIFIER-MONTANT-TRANSFERT

Montant = (UN PARAMETRE VALEUR = (UN Montant-TRANSFERT)

FAIRE EN SEQ :

(OPTIONNELLEMENT (CHERCHER ZONE-Message)

(LIRE (Message) DANS ZONE-Message))

(CHERCHER (Sasie-Montant) DANS ECRAN-Tansfère)

(CLAVIER : Taper les numéros du clavier correspondants au montant désiré)

(AFFICHER(Montant) DANS ZONE-Saisie-Montant)

(LIRE (Montant) DANS ZONE-Saisie-Montant)

(DOIGT : " placer-doigt-sur " (ICONE de Confirmation) "rester pendant au moins(0.20sec) ")

CLG « Caisse Express » (23)

- Pour mieux détailler une description, il faudra encore décrire les dispositifs physiques d'entrée et de sortie, pour bien spécifier les éléments physiques qui sont en contact avec l'utilisateur. Le modèle CLG, par contre, ne fournit aucun formalisme pour cette partie, ce que limite donc, une description bien détaillée de l'interface.

CLG « Caisse Express » (24)

- L'utilisation de CLG comme aide à la conception nous permet d'évaluer cette conception sans la participation de l'utilisateur. La structure théorique de la conception CLG est organisée en catégories (les différents niveaux), qui est semblable à celle selon laquelle que l'utilisateur organise la connaissance qu'il a du système [COU 90]. CLG propose donc un processus descendant "top down" de conception dans lequel est d'abord spécifié le modèle conceptuel du système et ensuite, un langage de commande avec lequel l'utilisateur entre en contact avec le système. Cette structure nous permet d'évaluer la prise en compte de l'aspect humain lors de la conception d'une interface, en regardant :
 - la façon que les tâches ont été décomposées,
 - les méthodes d'accomplissement des tâches,
 - les éléments d'interaction entre le système et l'utilisateur,
 - l'ordre des spécifications et les temps d'interaction.

CLG « Caisse Express » (25)

- L'utilité de l'interface peut être évaluée en regardant les services offerts par le système aux utilisateurs, conçus lors de cette étape, et les services présents dans la spécification des besoins. La qualité de l'interaction, déterminée par les décisions prises par le concepteur à chaque changement de niveau, doit être évaluée par rapport aux règles ergonomiques, car CLG ne fournit aucune aide au choix des décisions conceptuelles. Par contre, cette conception, décrite à l'aide d'un langage formel, peut être évaluée sur un outil qui reconnaît le langage CLG, et qui applique les règles ergonomiques.
- Les limites de CLG, déjà mentionnées précédemment, sont donc l'absence d'aide à la structuration et de moyens d'expression du parallélisme et des interruptions. Dans notre application, les interruptions sont représentées par les traitements d'erreurs, par exemple la spécification d'un montant de transfert invalide, d'un numéro de compte invalide, d'un service impossible de réaliser. Dans ces cas, nous n'avons aucune information de comment l'interface doit procéder. Notre application n'a pas de tâches parallèles (au moins entre les tâches fournies par le système).

Comparaison entre CLG et GOMS

- Contrairement au modèle GOMS, le modèle CLG ne fournit pas des règles de sélection de la méthode d'interaction.
- La structure de CLG est plus liée à l'aspect fonctionnel tandis que celle de GOMS est liée à l'aspect temporel (temps mis pour réaliser une tâche), ce qui fait de GOMS un modèle purement quantitatif, sans aucune référence à un modèle conceptuel.
- GOMS sert donc plutôt à l'évaluation de la performance, pendant que CLG nous donne aussi une analyse de la sémantique du système, du point de vue de l'utilisateur.