Interaction multimodale

Jean Caelen
CLIPS-IMAG
Jean.Caelen@imag.fr

1. Interaction homme-machine

Le terme « communication homme-machine » semble abusif : la machine n'est pas un être *social*, n'a pas d'intention ni de culture. Elle ne peut pas agir non plus sur le monde réel (on ne peut pas lui dire : « peux-tu fermer la porte s'il te plaît ? »). Elle n'a de prise que sur son propre monde. Le terme interaction homme-machine est plus adéquat.

La machine procure des outils pour réaliser une tâche, elle rend perceptible des objets virtuels, etc. Elle se présente donc comme un *interacteur*. Elle fournit un espace de travail, des outils et des méthodes. Mais pour tout cela elle doit être adaptée à sa tâche et / ou s'adapter à des tâches nouvelles, adopter un comportement « compréhensible », se montrer « conviviale », etc. Mais le paradoxe est évident : elle doit être *quelque peu sociale* pour collaborer efficacement avec un utilisateur autour des tâches de plus en plus complexes qui lui sont confiées.

L'interaction homme-machine se situe dans une relation opérateur-tâche où la machine à un rôle collaboratif [Falzon, 92]. Pour cela elle doit avoir des capacités qui lui permettent de comprendre les processus actionnels et dialogiques, c'est-à-dire elle doit posséder :
— la connaissance de l'opérateur,
— la connaissance du domaine de la tâche,
— des représentations d'elle-même (pour s'adapter),
— les règles de l'intervention pédagogique (aides, guides, exemples),
— les règles du dialogue (principes de négociation, de coopération, de réactivité, etc.),
— des règles de comportement "social",

et bien sûr tous les processus inférentiels mettant en œuvre ces connaissances. Ceci peut se représenter schématiquement par un modèle (fig. 1), dans lequel la machine, partant des actes produits par son interlocuteur humain, tente de les comprendre en les replaçant dans un cadre actionnel et dialogique (pseudo-social) pour générer des réponses sous

forme d'actions après avoir planifié ses réponses en fonction des contraintes interactionnelles.

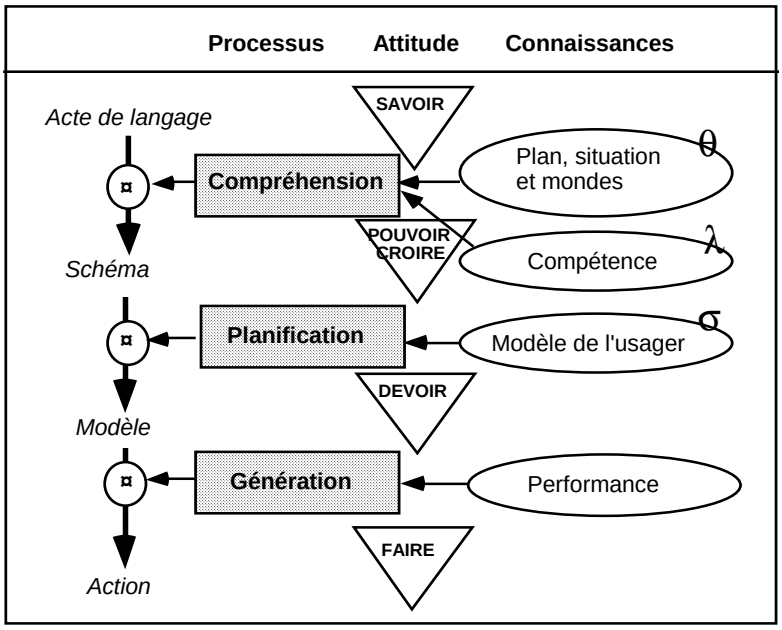


Fig. 1 : les processus inférentiels que doit posséder la machine. Un acte de langage est interprété en regard de la situation, essentiellement le plan d'action, but de la communication avec la machine. Cet acte est ensuite projeté dans un modèle par rapport auquel il est défini en "compréhension" pour finalement provoquer une action selon les performances de la machine manifestées dans une composante de "génération".

Acte de langage = est un acte multimodal répondant à la définition d'acte de langage c'est-à-dire ayant trois composantes, locutoire, illocutoire et perlocutoire,

Schéma = plan partiel induit des actes de langage analysés sous l'angle de la compétence linguistique et du contexte actionnel (plan, situation et mondes),

Modèle = séquence de scripts planifiés et sélectionnés en fonction des connaissances sur l'usager et des règles dialogiques,

Action = réponse de la machine (éventuellement multimodale) en terme de changement d'état dans la situation et dans les connaissances.

2. Interaction et interface : composants

Un système d'interaction homme-machine utilise des connaissances que l'on peut classer de la manière suivante :

2.1. Connaissances statiques

2.1.1. modèle de langage naturel

- composante reconnaissance: lexique, syntaxe, sémantique,
- composante génération: -idem-

Ces connaissances dépendent de l'application envisagée. Cependant il y a dans le lexique, une partie invariante, ce sont les mots-outils (articles, conjonctions, prépositions, etc.)

2.1.2. modèle de la tâche

- composante pragmatique: description des objets et de leurs relations relativement à l'application. *On emploie généralement des structures objets.*

- buts et sous-buts: chemins d'accès aux données et aux fonctions et la typologie des tâches. *On emploie ici aussi des structures objets pour définir les tâches et des graphes de dépendance pour décrire l'ordonnancement des tâches de l'application*

2.1.3. modèle du dialogue

description des diverses situations de dialogue par des scripts ou des scénarios

2.2. Connaissances dynamiques

2.2.1. modèle de l'utilisateur

- droits d'accès au système, privilèges, etc.

- connaissances de la machine sur l'utilisateur. *On utilise souvent la logique des croyances dans le cadre de la théorie des intentions.*

2.2.2. univers de la tâche

- base de faits ou de travail, historique des tâches et des objets de l'univers. *Cette base peut être tenue à jour par l'application elle-même.*

2.2.3. historique du dialogue

- à court terme
- à long terme.

C'est donc un **système basé connaissances** en frontal avec l'application (Fig. 2).

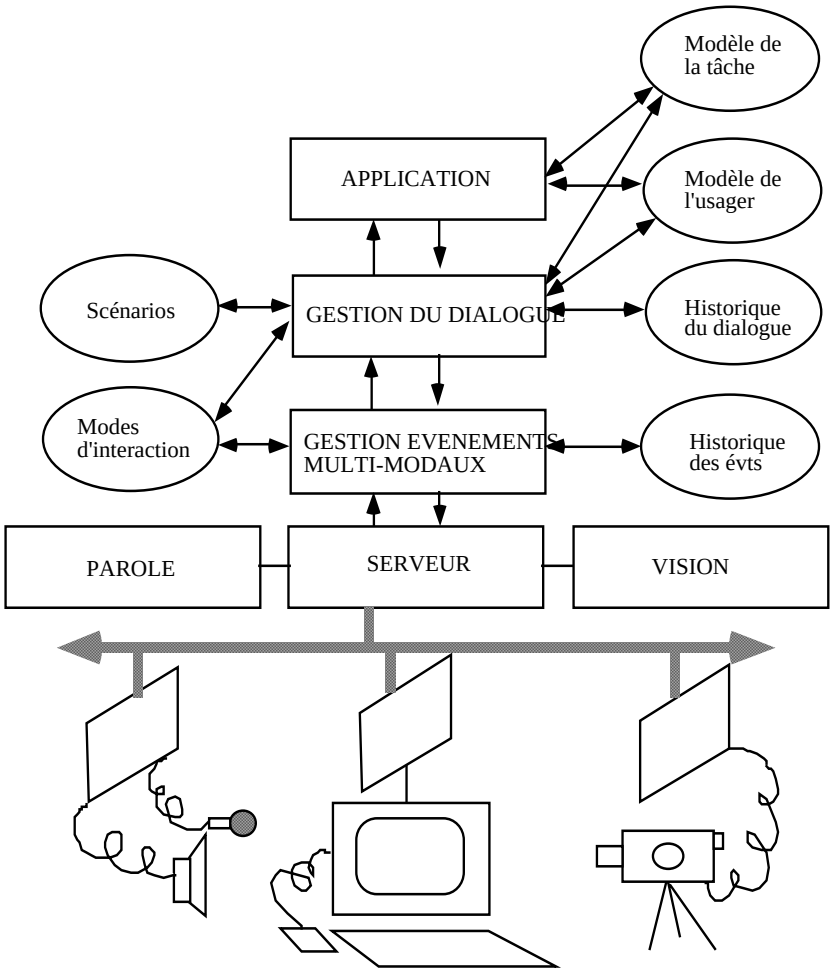


Fig. 2 : Architecture générale d'un système d'interaction multimodal (cas parole + geste + vision).

Les fonctions de la composante d'interaction sont :

1. gestion du canal de communication
- traitement du début de l'échange, de la fin, des relances
- mises en attente, reprises du dialogue

2. interprétation contextuelle
- ce module doit compenser certaines limites de la reconnaissance de la parole ou visuelle et en particulier doit raisonner à propos des
  + messages non reconnus ou inintelligibles
  + messages incomplets
  + messages ambigus
- les contraintes pragmatiques doivent être mises en jeu à la fois pour contraindre la compréhension (en proposition) et pour se focaliser sur la bonne interprétation (vérification d'hypothèses sur l'univers de l'application relativement à la tâche en cours)

3. gestion de l'interaction
- contestation, confirmation, reformulation

- accusé de réception,
- interrogation, demande d'aide
- détection des incohérences de la demande et formulation d'une réponse coopérative pour maintenir un dialogue constructif

3. Ergonomie cognitive de l'interaction

La finalité des études portant sur l'interaction homme-machine est la conception d'interfaces adaptées à l'activité des utilisateurs. Ce qui apparaît généralement à l'utilisateur au niveau de l'interface est le reflet de la structure des données et des tâches en machine. Celui-ci a, en effet, une activité organisée soit par sa formation, son expérience de la tâche, la pratique, son savoir-faire et les objectifs qu'il doit atteindre : dans une certaine mesure son activité est planifiée, il a une intention de départ qu'il réorganise en fonction des contraintes de la machine.

Le comportement : modèle de Rasmussen

Ce modèle décompose l'activité en trois types de comportement :
- le comportement fondé sur des habiletés (la conduite de véhicules par ex.),
- le comportement fondé sur des règles (apprises, acquises par l'expérience, etc.) pour des situations connues,
- le comportement fondé sur la connaissance ou raisonnement devant une situation inconnue.

L'ergonomie cognitive a pour domaine l'ensemble des activités mentales de sujets engagés dans une tâche qui correspondent donc aux niveaux 2 et 3 du modèle ci-dessus qui seules, sont verbalisables par l'utilisateur lui-même.

Les langages d'interaction en langue naturelle restreinte

La conception d'un dialecte dérivé de la langue naturelle (plutôt qu'un sous-langage ou qu'un langage formel) est la meilleure solution :
- pour faciliter l'apprentissage des entités et des opérations par l'utilisateur,
- au niveau de la machine car le lexique est bien défini et la syntaxe limitée.

Dans les langages opératifs homme-homme il n'y a pratiquement pas de syntaxe le vocabulaire est limité mais très spécialisé. Ce langage est très lié à la nature de l'application.

Devant une machine les utilisateurs "s'adaptent" en rendant leurs énoncés plus clairs : moins d'ellipses et d'anaphores, syntaxe plus souvent correcte (même si on ne leur demande pas). Pour la prosodie on a pu se rendre compte d'un phénomène analogue [Caelen-Haumont, 79].

Thomas propose de classer les utilisateurs en quatre groupes sur les deux paramètres = connaissance de l'informatique x connaissance du domaine. La production verbale se dégrade avec la charge de travail ou la concentration sur l'objectif.

Aspects linguistiques

Taille du vocabulaire : Par exemple, le langage de commande d'un éditeur graphique ne s'élève qu'à 189 mots (Hauptman et Green, 1983). Dans la plupart des applications on peut donc se contenter d'un nombre de mots assez réduit. Cependant il ne faut pas confondre ceci avec le nombre de mots à stocker dans le lexique du modèle de reconnaissance puisqu'il faut ici toutes les formes lexicales utiles (formes conjuguées, formes accordées, expressions, etc.)

Critères de choix du vocabulaire-noyau : La fréquence des mots, leur banalité et leur occurrence dans des expressions différentes. Le vocabulaire comporte toujours des mots rares et spécifiques.

La syntaxe : souvent la forme impérative ou impersonnelle, la syntaxe peut être restreinte (par ex. 14 règles pour Hendler et Michaelis, 1975), en phrases courtes comportant peu de références pronominales ou elliptiques, de métaphores et de métonymies. Par contre les groupes nominaux peuvent être riches (le petit livre rouge sur la table de gauche).

La sémantique : est surtout caractérisée par la monosémie lexicale. La sémantique est orientée par les objectifs.

La compréhension opérative

Est limitée au contexte de l'application. Elle nécessite un filtrage de certaines parties du discours (sans analyse syntaxique) puis une analyse syntaxique approfondie des parties sélectionnées. La compréhension est à dominance sémantique avec une stratégie descendante. Luzzati, dans DIALORS, ne modélise pas la syntaxe (on remplit seulement les attributs de schémas évoqués par des mots déclencheurs en parcourant la forme de surface de l'énoncé). Cependant on atteint vite les limites du système par cette technique. Il faut donc une stratégie plus complexe dans laquelle analyse globale et analyse de détail se complètent.

Les scripts

Sont actuellement parmi les sujets de recherche, comme voie d'amélioration possible des mécanismes de compréhension. Un script est un plan de schémas qui est évoqué dès que des conditions particulières sur la situation sont réunies.

4. Le dialogue multimodal

On fera la distinction entre multimédia et multimodalité. Le premier désigne les supports ou les véhicules de l'information le deuxième la substance de l'information :

média : microphone, écran, clavier, souris, caméra, etc.

modalité : parole, vision, écriture, geste, etc.

Complémentarité et coopération des média

L'interaction homme-machine doit s'appuyer sur une ergonomie d'harmonisation des moyens de communication que sont écran, clavier, souris, voix, image, etc. Par exemple, considérons la commande *"déplacer la fenêtre active vers la gauche"* : deux cas se présentent (a) soit il s'agit de déplacer une fenêtre sur une position précise —un moyen de pointage comme la souris est alors indispensable— (b) soit il s'agit simplement de dégager un espace invisible et le positionnement précis de la fenêtre à déplacer n'est plus nécessaire, auquel cas un ordre oral est plus efficace puisqu'on continue à travailler "mains occupées". Cet exemple montre qu'il n'y a pas équivalence entre une action "souris" et une action "voix" mais qu'elles se complètent en entrant dans des champs d'action et d'utilisation spécifiques. C'est encore plus vrai lorsque l'on dit *"pousse la fenêtre ici"* en désignant la position voulue par la souris.

De manière générale, il vaut mieux entrer des données --nombres, noms (de fichiers par ex.)-- au clavier (pour des raisons de fiabilité et de taille de vocabulaire), les opérations de mouvements fins --réglage de taille de fenêtre, déplacements, pointage, etc.-- à la souris et ne garder pour la communication orale que des commandes de niveau élevé, par exemple "*ouvrir un fichier sur le lecteur interne* " équivalente à une longue séquence de "clics" sur les menus.

Dans le cas de la réponse orale —pour des messages d'aide, de demande de confirmation ou de renseignements complémentaires, etc.— le problème est exactement symétrique : certains messages sont mieux captés par l'oral que par le texte écrit (messages d'alerte notamment, commentaires, aides).

Le modèle de dialogue

Littéralement le "dialogue" sous-tend un fonctionnement de type conversation c'est-à-dire une intervention alternée entre l'homme et la machine (et de ce fait, souvent guidée par la machine). Dans une interface multimodale, où les objets sont "vus" à l'écran il est préférable de généraliser pour le dialogue la notion de boucle d'attente sur événements qui donne l'impression à l'utilisateur qu'il agit en "maître" sur l'univers de l'application, (comme dans le cas de menus). Dans le concept de "manipulation directe" on agit directement sur le modèle d'univers présenté par le système (du côté ergonomique le dialogue parait moins abstrait car l'utilisateur n'a pas besoin de mémoriser tout le vocabulaire utile à la communication). L'univers est un "univers sémantique" dans lequel le dialogue devient un ensemble de sous dialogues entre l'utilisateur et les divers objets de l'univers. Ces objets peuvent être décrits avec des formalismes centrés-objets.

Le dialogue sous-tend cependant des parties "dirigées par la machine" dans les phases

suivantes :

- phase d'introduction, la machine pose des questions pour cerner le type d'usager qui veut utiliser l'interface. A l'aide de critères de compétence multiples comme *compétence en informatique, habituation aux interfaces, compétence dans le domaine traité par l'application,* etc., la machine se fait une "idée" de l'utilisateur (logique des croyances),
- phase d'aide intelligente et personnalisée,
- phase d'entrée d'informations complémentaires demandées par la machine pour exécuter correctement une tâche,
- phase de traitement des erreurs ou des ambiguïtés,
- phase de conclusion pour sauvegarder et gérer des fichiers, voire préparer la future séance interactive, etc.

La fig. 4 montre un script général de dialogue possible.

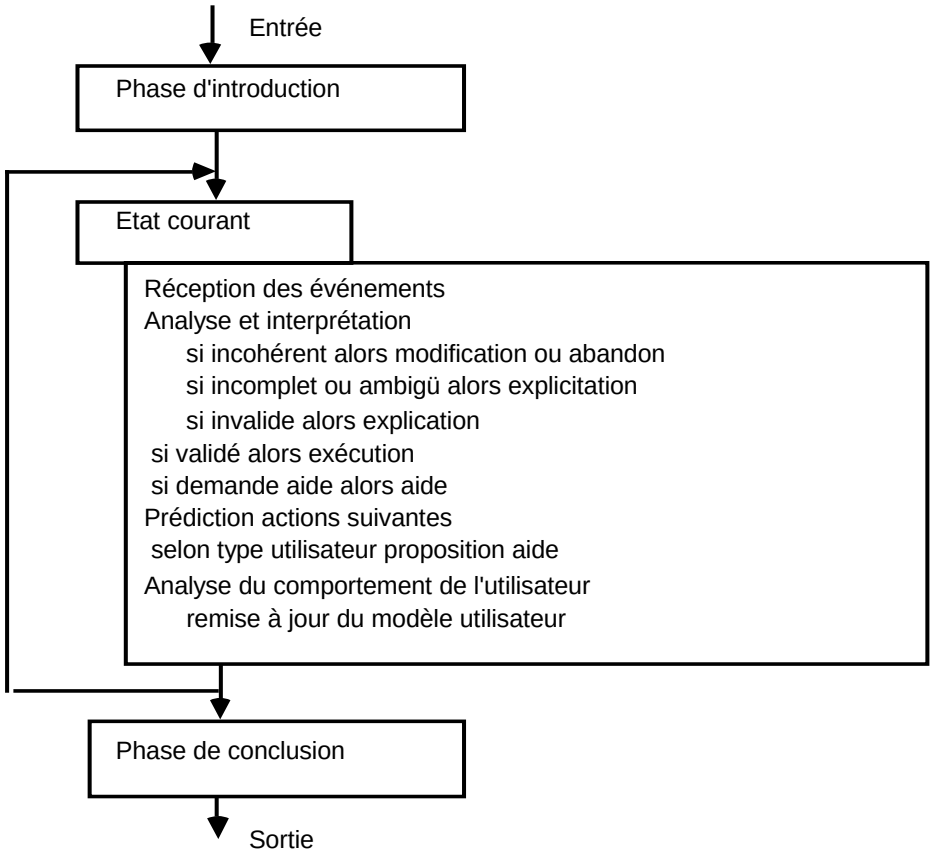


Fig. 4 : un script de dialogue général dans lequel les phases d'introduction et de conclusion sont dirigées par la machine, tandis que l'état courant est dirigé par l'utilisateur et un script de dialogue particulier pour tenter d'exécuter une tâche.

Les niveaux de langages

On peut distinguer deux niveaux de langage liés l'un **(L1)** au système d'exploitation et/ou au gestionnaire graphique (ouverture de fichiers, déplacements de fenêtres, navigation dans les menus, etc.) c'est-à-dire à l'interface homme-machine et l'autre **(L2)** à l'application. Le premier de ces niveaux reste relativement indépendant de l'application

puisqu'il concerne surtout l'interface. Par contre pour le second, chaque application ayant son vocabulaire et sa syntaxe propres, une mise en œuvre devient obligatoire.

On peut dresser une typologie des applications :

A) Manipulation d'objets d'un univers (logiciel de dessin, CAO, etc.) : ici il n'y a pas de but à long terme mais une série de tâches de détail à exécuter : la machine ne peut percevoir qu'une certaine intention à court terme de l'utilisateur et doit contrôler a posteriori les "manoeuvres" effectuées. La prédiction est faible dans le dialogue, les guides ne peuvent être que de vagues suggestions. Le dialogue est **dirigé par les objets**.

B) Tâches planifiées (calcul, saisie, visualisation, etc.) : beaucoup de logiciels fonctionnent sur le principe suivant : pour atteindre un résultat les tâches doivent être ordonnancées. Le dialogue est alors **dirigé par la tâche** : faire A1 puis A2, si A2 échoue alors faire A3 puis refaire A2, sinon faire A4, etc. Les intentions de l'utilisateur sont claires, il doit atteindre un résultat à l'aide d'une méthode décomposée en étapes en un minimum de temps. Le dialogue doit viser à clarifier le cheminement de l'utilisateur dans le dédale des possibilités offertes par le logiciel et lui donner les moyens d'y parvenir : saisie des paramètres convenables, choix des méthodes les plus efficaces, planification correcte des étapes, etc.

C) Consultation et renseignement (bases de données, services, etc.) : ici l'utilisateur ne sait pas trop ce qu'il cherche, ni comment l'obtenir. Il a des difficultés à formaliser sa démarche. La machine doit alors faire de grands efforts de compréhension, le dialogue doit être **dirigé par le but**.

Généralement, dans un dialogue de manipulation d'objets, le langage utilisé est opératif: le vocabulaire est limité, la syntaxe peut être négligée (style abrégé) et la compréhension peut être dirigée par des schémas. Le premier mot de la phrase (souvent un verbe) sert de déclencheur à un schéma et les mots suivants permettent d'orienter la particularisation. Certains incidents de communication peuvent être mis en relation avec le caractère inaproprié du premier mot qui peut orienter sur un schéma incorrect. C'est le cas par exemple, quand le mot déclencheur n'est pas en tête du message ou quand ce mot est polysémique. Ces problèmes doivent donc être pris en compte dans le modèle de la tâche.

Dans un dialogue d'interrogation de bases de données les phénomènes linguistiques sont beaucoup plus complexes. A travers la forme de surface de la demande il faut souvent détecter l'intention de l'utilisateur.

L'aide intelligente

Il faut fournir pour l'intégration du dialogue des outils inspirés de l'EIAO (situation où la machine est la plus coopérante) pour lesquels les aides fournies sont de type explication, exemples, guidage, etc. Ces aides sont activées automatiquement (ou à la suite de demandes de l'utilisateur) par des démons attachés aux scripts de dialogue. Ces démons analysent la fréquence des retours arrière, des tâtonnements, des hésitations, des incohérences de l'usager au cours de l'utilisation de l'interface. Selon le type de démon

activé, et d'après le modèle attaché à l'usager, une aide personnalisée pourra être proposée.

5. L'interface homme-machine (IHM)

Le dialogue doit être intégré dans l'Interface homme-machine mais ne peut être complètement désolidarisé de l'application. Une certaine duplication des informations entre l'application et le dialogue doit même être envisagée.

Le modèle adopté devra contenir au moins pour la parole :

- une base des objets de l'univers (lexique) avec leurs attributs sémantiques propres à(ux) l'application(s),
- une base de tâches associée à la liste des verbes correspondants et à leurs relations de dépendance,
- un historique des actions exécutées pour résoudre le problème des anaphores et des ellipses au cours du dialogue,
- une mémoire partagée avec le logiciel de reconnaissance et de synthèse de la parole,
- un moteur d'inférences propre à résoudre certains problèmes de raisonnement liés au traitement du langage naturel. Il est évident qu'il s'agit ici de développer (ou d'utiliser) un module de compréhension de la parole adapté aux applications envisagées. Il doit permettre en particulier de générer des phrases variées à partir d'un lexique afin de synthétiser des messages qui ne soient pas trop récurrents dans le temps (pour éviter le phénomène de lassitude de l'auditeur),
- un "speech manager" pour gérer les événements parole de bas niveau.

Le modèle Seeheim

Ce modèle est de type série (Fig. 5). Son UIMS (User Interface Management System) se décompose en trois parties :

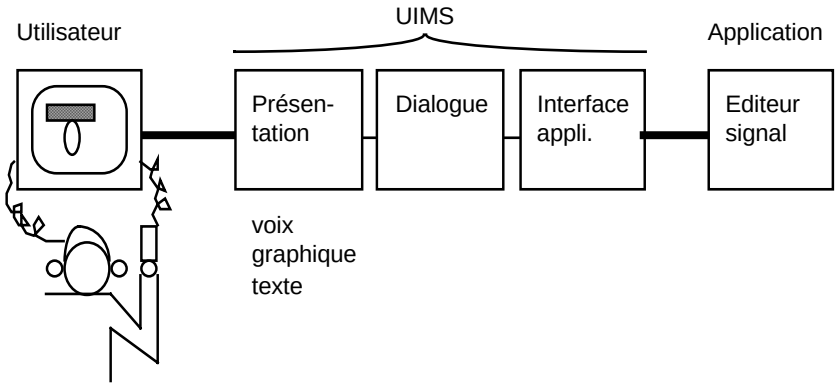


Fig. 5 : Le modèle d'UIMS Seeheim

Présentation:

Graphique : boutons, icônes, ascenseurs, flèches

Sonore : parole, motifs musicaux
Données d'entrée : clics souris et mouvements, textes, parole
Réponse (feedback) : vidéo inverse, action demandée, message vocal,
contrôle lexical

Dialogue:
reconnaît les requêtes, les prépare pour l'interface de l'application, connaît l'état de la situation
contrôle syntaxique

Interface de l'application:
pont entre l'application et les autres interfaces,
contrôle sémantique

Le modèle centré-objet, le modèle PAC

PAC = Présentation Abstraction Contrôle [J. Coutaz]

C'est un modèle orienté objet dans lequel les fonctions décrites dans le modèle Seeheim sont moins distinctes et donc dans lequel les fonctions syntaxiques et sémantiques sont mieux intégrées. On y distingue aussi trois parties:

Présentation : contrôle syntaxique vis-à-vis de l'utilisateur
Abstraction : fonctions ou attributs fonctionnels sur les objets
Contrôle : gestion des liens entre Présentation et Abstraction

Présentation:
graphique: boutons, icônes, ascenseurs, flèches
sonore: parole, motifs musicaux
données d'entrée: clics souris et mouvements, textes, parole
réponse (feedback): vidéo inverse, action demandée, message vocal, représentation syntaxico-sémantique liée à l'application

Contrôle:
reconnaît les requêtes, les prépare pour l'interface de l'application, connaît l'état de la situation

Abstraction:
connaissances syntaxico-sémantiques liées au modèle abstrait représenté par des schémas (d'objets et de tâches)

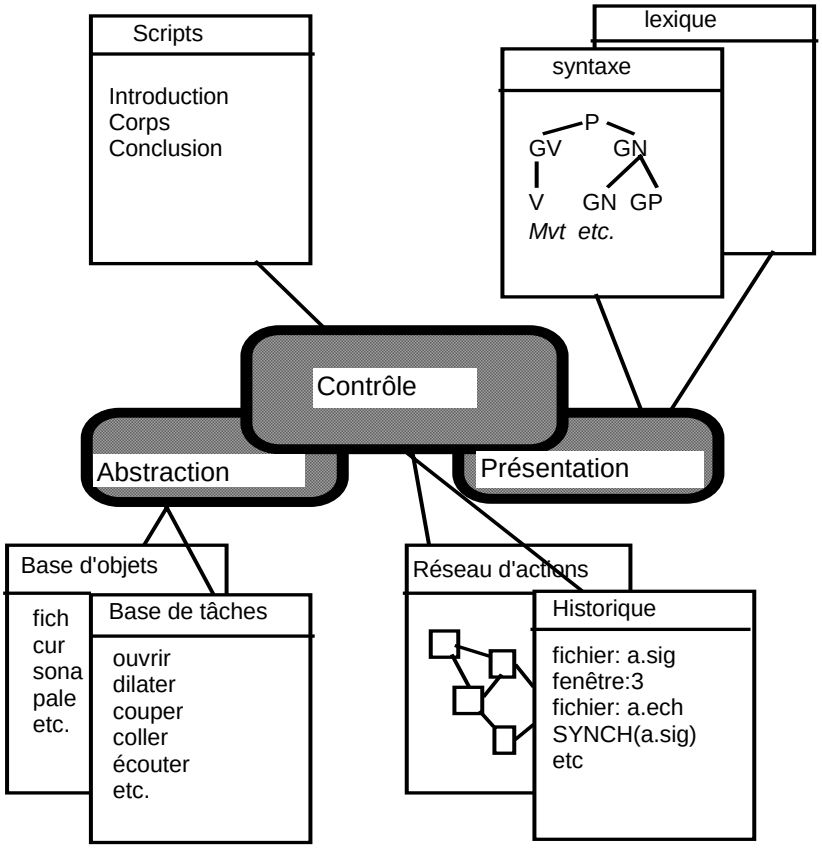


Fig. 6 : Les agents de plus haut niveau pour la composante orale dans une architecture PAC.

Les vues sont les diverses formes de présentation des objets (voix, graphiques, etc.) et sont décrites comme des objets qui permettent d'instancier les nœuds correspondant aux données de l'application (le comment et le quoi).

Les plans de présentation sont des procédures qui permettent de présenter une instance. Ces plans doivent être interprétés (ils sont écrits en P-code sur Mac et Sun qui l'intègre dans leur DPU: Display Processor to Draw)

Le feedback est une notion très importante en dialogue. Il existe trois niveaux dans la portée d'une réponse

lexical: réponse de bas niveau: ex. on clique dans un bouton il passe en vidéo inverse (sans analyse de l'action correspondante)

syntaxique: l'action est-elle compatible avec les précédentes ? ex. on éteint les menus non activables à un moment donné

sémantique: l'action est-elle totalement définie et a-t-elle un sens ? ex. on tente de faire un zoom sur une section de sonagramme trop courte.

C'est cette notion de feedback sémantique qui amène à intégrer davantage l'UIMS dans l'application

Le undo est une deuxième notion importante: récupérer les mauvaises commandes ce qui oblige à faire des back up fréquents. Il faut définir ces points de back up dans le dialogue pour arriver à un système tolérant aux erreurs de commande. Il y a deux techniques: on mémorise l'état avant la commande ou on calcule, lorsque c'est possible, la commande inverse.

Technique: la base d'objets peut être décrite par une grammaire d'attributs et compilée pour obtenir le graphe de dépendance des attributs, objets, actions, etc. A la suite de cela l'algorithme de Reps permet de faire une analyse syntactico-sémantique incrémentale des objets.

Indépendance entre IHM et dialogue ?

On peut distinguer deux types d'interaction homme-machine, l'une uniquement au niveau de l'interface et/ou du système (navigation dans des menus, commandes système, etc.), l'autre au niveau de l'application (commandes, interrogations sur les objets mêmes de l'application). Les architectures peuvent différer grandement entre ces deux types d'interaction.

(a) Dialogue au niveau de l'interface (ex. "Journal")

Niveau O : équivalence clavier-voix : le menu est déroulé il s'agit de choisir une rubrique visible. Pour la navigation par la voix, énoncer un digit est préférable à énoncer le mot de commande de la rubrique choisie car il n'y a pas d'apprentissage linguistique ni acoustique et pas de charge mnémonique pour l'utilisateur. Un fichier de ressources définissant les dépendances entre les rubriques suffit pour lier le lexique de digits aux rubriques. Cela ne nécessite pas de véritable gestion de dialogue.

Niveau O+ : navigation sans contrainte : dans le cas précédent le déroulement des menus est arborescent. On peut envisager, par la voix, de créer des commandes multiples (choix de plusieurs rubriques comme "ouvrir un nouveau fichier et coller") ou de naviguer dans les menus dans un ordre quelconque et sans visibilité ou de créer des abrégés vocaux. Il y a ici complémentarité entre clavier, voix et souris. Une gestion de dialogue type peut être faite, valable pour toutes les applications à menus —puisque seule l'interface est concernée— dont il faut simplement adapter le lexique et la syntaxe.

Dans les deux cas les messages vocaux de sortie sont en nombre limité et fixes —éventuellement certains blancs peuvent être complétés par des variables— ils peuvent être stockés en parole compressée.

Supposons que l'interface de l'application se présente sous forme de menus comme celui-ci (type MacIntosh™) (Fig. 7) :

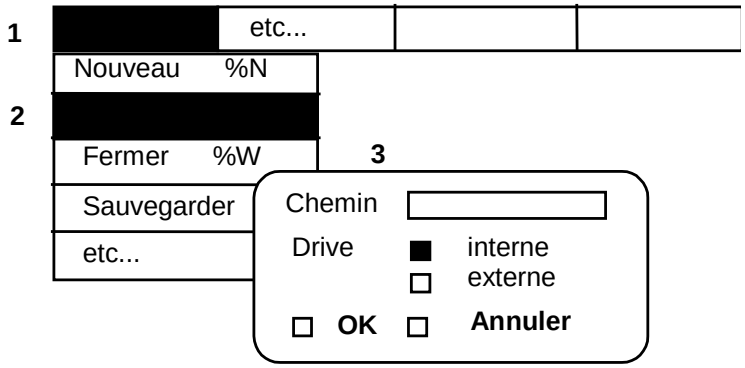
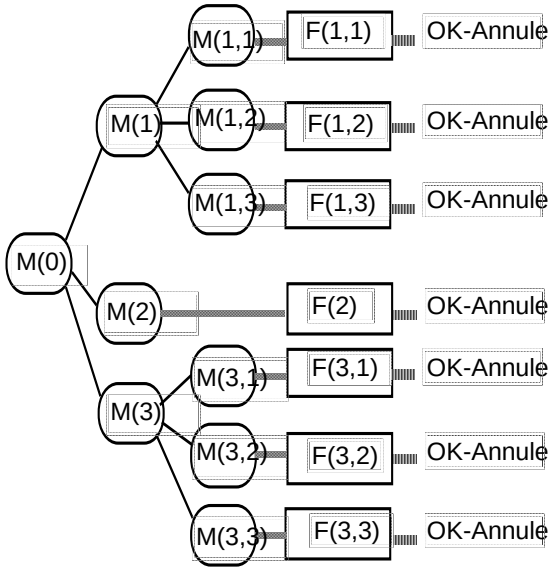


Fig. 7 : Résultat d'affichage de menus (1=barre fixe, 2=menu déroulant) et formulaire (3) pour lequel il reste encore à entrer l'information "chemin" puis à valider la commande.

Cet affichage résulte indifféremment d'une succession d'événements tels que :
(a) exclusivement souris = k(fichier)+k(ouvrir)+k(Drive=interne) ou
(b) exclusivement parole = "ouvrir un fichier sur le drive interne" ou
(c) de toute autre combinaison comme
k(fichier)+"ouvrir"+ k(Drive=interne)
%O + "Drive interne" etc.
avec les conventions d'écriture k(x) clic sur le menu x, "y" ordre entré par la voix et %z abréviation clavier.

A la suite de cet affichage d'autres paramètres restent à définir comme CHEMIN (puis OK ou ANNULER), dont la valeur peut à son tour, être entrée au clavier ou prononcée voire épelée (déconseillé cependant). Plus généralement la structure des menus est arborescente et les feuilles sont reliées à des formulaires (éventuellement vides). Ainsi les nœuds représentent l'action ou la suite d'actions à effectuer —c'est le verbe dans le cas de la parole— mais aussi l'objet sur lequel porte l'action —le GN— une modalité de l'action —les circonstants— ou même une méta-action. Les formulaires représentent les arguments des fonctions associées aux actions —ce sont les GN, GP, etc— choisis en liste fermée ou en liste ouverte. On a donc la structure formelle suivante :



où M(i,j,k) désigne un menu et F(i,j,k) le formulaire associé. En général M(0) est la racine abstraite (point d'entrée-sortie de l'application), M(i) est la barre de menu fixe et M(i,j,k) un menu de profondeur 3 (il n'y a pas plus de trois niveaux d'imbrication pour des raisons d'ergonomie).

```
M(0) : char 'Racine'
    M(1) : char 'Fichier'
        M(1,1) : char 'Nouveau'
            clavier : %N
            parole : "Ouvrir un nouveau fichier"
        M(1,2) : char 'Ouvrir'
            clavier : %O
            parole : "Ouvrir" DET "fichier"
        M(1,3) : char 'Fermer'
            clavier : %W
            parole : "Fermer" DET($Def) "fichier"
        M(1,4) : char 'Sauvegarder'
            clavier : %S
            parole : "Sauver"|$Syn DET($Def) "fichier"
        M(1,5) : etc.
    M(2) : etc.

F(1,2) : sorte-de Formulaire
    $Chemin = char(20)
    $Drive = [interne, externe]
    Validation = [OK, Annuler]
```

avec la notation "Sauver"|$Syn DET($Def) "fichier" signifiant : verbe "sauver" ou un synonyme suivi d'un déterminant défini et du nom "fichier".

Niveau 1 : Dialogue au niveau de l'application (ex. ICPdraw)

Si l'on veut détacher la gestion du dialogue de l'application et l'insérer dans l'interface il faut que les objets de l'application soient connus et correctement représentés dans le module de gestion de dialogue. Cela entraine une certaine duplication des informations.

8. Interaction multimodale

Une interface homme-machine (IHM) multimodale dispose de plusieurs modes d'entrée et de sortie. Ces modes correspondent à certaines des modalités sensorielles et motrices de l'humain. Les problèmes qui distinguent les interfaces multimodales des interfaces classiques naissent de :

• La gestion des modes aux niveaux [Bourguet et al., 92]
  — des événements (chronologie, synchronie)
  — des informations (unités, actes)
  — et du contexte interactionnel

• La fusion / fission des informations au niveaux
  — morphosyntaxique
  — sémantique et / ou pragmatique (résolution de la coréférence)
  — actionnel (intégration de la multimodalité au niveau de la couche interaction / dialogue)

• L'échange des informations avec les autres modules de l'interface et le noyau fonctionnel de l'application.

A chaque mode, est associé un modèle de représentation des informations qu'il véhicule. Ce modèle dépend de la granularité des événements de bas niveau sur laquelle il est construit. Ainsi pour un "geste" le système délivre des vecteurs de coordonnées de points dans le temps alors que pour la parole ce sont des chaînes de caractères correspondant à des mots ou des phrases reconnues ou bien du son échantillonné. Les fréquences d'échantillonnage de ces données sont différentes d'un média à l'autre. Les problèmes qui se posent dans une interface multimodale sont donc :

(a) l'acquisition des signaux fournis par l'usager,
(b) leur reconnaissance automatique,
(c) la compréhension des signes qu'ils véhiculent,
(d) leur interprétation coréférentielle,
(e) la construction d'un message actionnel multimodal.

Le cheminement des informations passe par une mise en forme, une représentation abstraite, une fusion et enfin une transmission à la couche « dialogue » [Taylor, 89] qui se trouve de fait posé au niveau le plus haut.

8.1. La gestion des modes

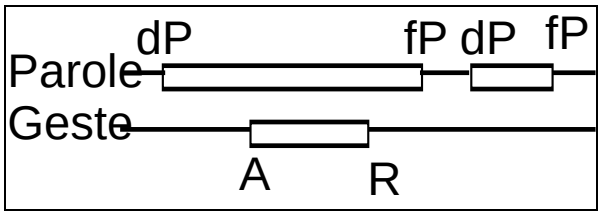
La gestion des modes est une opération qui consiste à :

- capter les événements en provenance des serveurs de médias (inversement à émettre pour les sorties),
- construire les structures événementielles et informationnelles,
- gérer le contexte interactionnel, en fonction du type d'information et des connaissances transmises par les niveaux adjacents (module de fusion, module de dialogue par exemple),
- maintenir un historique pour ce contexte,
- mettre à profit les connaissances sur l'usager au niveau sensori-moteur (temps de réaction, préférences modales, etc.).

Pour avancer clairement dans la problématique présentée ci-dessus, il est important de bien distinguer les événements (qui reflètent l'organisation physique des actes) des informations (ou unités qui les composent).

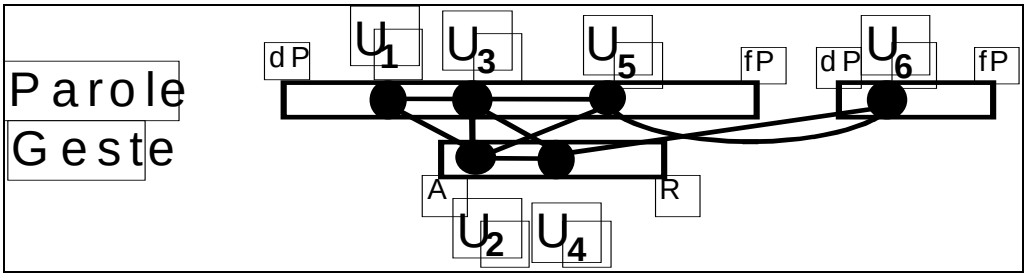
4.1.1. *Evénements, informations*

Définition d'un événement : un événement est un début, ou une fin d'un signal externe à la machine : il signale un changement perceptible sur un média. Cette définition est centrée sur la machine et non sur l'utilisateur, plus précisément sur les canaux d'entrée-sortie que nous appelons médias.



Définition d'une information : une information est une unité signifiante, mais qui ne prend pas la même signification pour l'usager et pour la machine. C'est,

- une unité sémiotique pour l'usager,
- une unité référentielle pour la machine.



Il est clair qu'il existe des relations sémantiques entre les unités et des relations temporelles entre les événements.

4.1.2. *Le contexte interactionnel*

Définition du contexte interactionnel : le contexte interactionnel est le triplet {usage des modes, dépendance des informations, animation}. Le premier attribut dénote l'usage (de facto les capacités du système) séquentiel ou parallèle des modes, le second l'indépendance des informations véhiculées sur les médias et le troisième la dynamique du monde c'est-à-dire les actions à effet continu et les actions à effet instantané. Nous ne nous intéresserons qu'aux deux premiers attributs qui définissent quatre contextes interactionnels : exclusif, concurrent, alterné et synergique {Caelen, 91], [Coutaz, 92].

Le contexte "Concurrent"

Il se définit par :
usage des modes : sans contraintes temporelle (parallélisme possible)
dépendance sémantique : pas de coréférence intermodale entre les unités,
$L\{u_{ij}(k), u_{i'j'}(l)\} = \emptyset$ pour $i \neq i'$.

Propriétés : L'anaphore est mal résolue lorsque la référence est portée par un autre mode, la déixis ne peut être résolue.

Le contexte "Alterné"

Il se définit par :
usage des modes : $Début_i(k) > Fin_{i'}(k-1)$ avec $i \neq i'$
dépendance sémantique : pas de contraintes coréférentielles

Propriétés : L'anaphore est bien résolu lorsque la référence est portée par un autre mode, la déixis peut être résolue. L'usage alterné des modes entraîne une lourdeur qui pénalise la coordination perceptive / motrice de l'usager.

Le contexte "Synergique"

Il se définit par :
usage des modes : aucune contrainte

dépendance sémantique : pas de contraintes coréférentielles

Propriétés : L'anaphore est bien résolue lorsque la référence est portée par un autre mode, la déixis également. L'usage synergique semble être la meilleure solution si l'on sait résoudre les problèmes coréférentiels intermodaux, c'est également le plus économique au niveau sensori-moteur. Mais elle pose problème pour traiter les anticipations ou les retards.

4.1.3. *Formalisation*

Si nous considérons un système multimodal le plus général possible, il n'y a pas lieu de le considérer entièrement centralisé ; supposons au contraire qu'il utilise des ressources délocalisées [Decouchant et al., 89], appelées serveurs de médias. Ces serveurs ont par exemple des cartes de reconnaissance ou de synthèse de la parole ou ne sont que des logiciels de reconnaissance de geste sans hardware particulier en dehors d'une souris. Le système multimodal devient alors lui-même un serveur sans média. Il peut être à son tour distribué. Ses fonctions sont de gérer les modes, événements et services, et de fusionner les informations jusqu'à un certain niveau pour les transmettre à un module de dialogue ou un collecticiel ou tout autre application.

Au niveau formel continuant à maintenir une claire distinction entre événements et informations, nous définissons :

<u>Les structures événementielles</u>

soit $\mu_i(k)$ = ième acte en mode k reçu (émis) par le système multimodal de (vers) un ensemble de serveurs $\{\Sigma\}$, on pose :

événement-de-acte : attaché-à $\mu_i(k)$
| type : $e_i(k) = \{d\mu_i(k), f\mu_i(k)\}$
| mode : k
| date : $\tau\ (e_i(k))$
| n°-ordre : i
| provenance / destination : $\{\Sigma\}$

soit $u_{ij}(k)$ = jème unité contenue dans $\mu_i(k)$, on pose :

événement-d'unité : attaché-à uij(k)
| type : $e_{ij}(k) = \{du_{ij}(k), fu_{ij}(k)\}$
| acte : $\mu_i(k)$
| date : $\tau(e_{ij}(k))$
| n°-ordre : j

<u>Les relations événementielles</u>

chronologique ($\leq$), monomodale

$e_{ij\text{-}p}(k) \leq e_{ij}(k)$ ssi $\forall p \geq 1$, $\tau(e_{ij\text{-}p}(k)) \leq \tau(e_{ij}(k))$

synchronique (≈), multimodale
$\forall k \neq k'$, $e_{ij}(k) \approx e_{i',j'}(k')$ ssi $e_{ij}(k) \in [du_{i',j'}(k'), fu_{i',j'}(k')]$ ou $e_{i',j'}(k') \in [du_{ij}(k), fu_{ij}(k)]$
avec,
$e_{ij}(k) \in [du_{i',j'}(k'), fu_{i',j'}(k')]$ ssi $\tau(du_{i',j'}(k')) \approx \tau(e_{ij}(k)) \approx \tau(fu_{i',j'}(k'))$

ces relations sont aussi applicables aux événements d'actes.

Propriétés : (≤) est une relation d'ordre partiel, (≈) est une relation d'équivalence

unités (actes) synchrones
deux unités (actes) sont synchrones s'ils possèdent deux événements synchrones
$\forall k \neq k'$, $u_{ij}(k) \approx u_{i',j'}(k')$ ssi $\exists\, e_{ij}(k) \approx e_{i',j'}(k')$ -id- pour les actes

la durée de deux unités (actes) synchrones est :
$\delta(u_{ij}(k) \approx u_{i',j'}(k')) = \max[\tau(e_{ij}(k)),\tau(e_{i',j'}(k'))] - \min[\tau(e_{ij}(k)),\tau(e_{i',j'}(k'))]$ -id- pour les actes

<u>Les deux définitions du "présent"</u>

- le *présent instantané* : durée de l'unité la plus courte à un instant donné

- l'*épaisseur du présent* : intervalle de temps défini par la durée de tous les actes synchrones à un instant donné. Cette épaisseur est variable au cours du temps.

Cas particuliers :
— dans un système alterné il n'y a pas d'unités ni d'actes synchrones,
— dans un système concurrent la gestion des modes s'effectue comme dans un système synergique mais il n'y a pas de niveau de fusion d'informations.

<u>Le contexte interactionnel (dans un système dynamique)</u>

Un système est dit dynamique s'il est capable de gérer différents contextes interactionnels. Le contexte interactionnel a été décrit ci-dessus. C'est le triplet $\mathbf{C_o}$ = {usage des modes, dépendance des informations, temporalité}

usage des modes : il est déterminé par la boucle action / perception et les contraintes mécaniques du système
ex. Mettre(Objet, Lieu)
"mets ça ici" < dg(ça) < dg(ici) => alterné
("mets ça ici" ≈ dg(ça)) < dg(ici) => synergique(p+)
("mets ça" ≈ dg(ça)) < ("ici" ≈ dg(ici)) => synergique

“mets” < (“ça” ≈ dg(ça)) < (“ici” ≈ dg(ici)) => synergique(g+)

avec,

“ ” = acte de parole
dg = acte de désignation gestuelle
p+ = dominance du mode parole
g+ = dominance du mode gestuel
dans le dernier cas le geste rythme la parole et la détermine temporellement. Les événements sont synchrones et les informations dépendantes ; on en déduit que le contexte interactionnel est synergique à dominance gestuelle.

dépendance sémantique : elle est déterminée par les relations sémantiques / pragmatiques entre les unités
ex. dg(triangle) ≈ “déplace le cercle” => concurrent
les deux actes sont synchrones et indépendants car l’objet désigné triangle ne coréfère pas avec l’objet cercle de l’acte de parole. On en déduit le contexte interactionnel “concurrent”.

Ces quelques exemples montrent que le contexte interactionnel se déduit de d’organisation et du contenu même des actes. Cela fait qu’il ne peut être déterminé que de manière inférentielle.

8.2. Fusion/fission des informations

Le problème central dans une interface homme-machine multimodale se situe dans la fusion (en entrée) et la fission (en sortie) des informations intermodales. Placé au-dessus de la gestion des modes, le module qui traite de la fusion (resp. fission) fait le lien avec le module qui traite du dialogue (voir §3).

Cerner les fonctions et les limites d’un module de fusion est chose délicate, car sa spécificité peut être contestée [Gaiffe et al., 91] : on pourrait en attribuer tous les rôles au contrôleur de dialogue qui analyserait les informations prélevées au bas niveau et se chargerait de la fusion des informations dans un processus englobant [Wilson et al., 91]. Quelles sont les raisons qui plaident en faveur d’un tel module distinct et spécifique pour les IHM multimodales ?

La discussion générale de cette question est vaste ; elle devrait porter sur les points suivants :

- Stratégies de fusion
  - Quand ?
    - au plus tôt (précoce)
    - au plus tard (différé)

par étapes
Comment ?
autour d'une structure commune
et d'un mode dominant
"grammaire" d'unification (langagière bien formée)
sans mode dominant
"grammaire" multimodale
par une théorie de l'action
sans structure commune
Où ?
centralisée dans le contrôleur de dialogue
de manière répartie et progressive
Avec quelle logique ?

• Critères de fusion
de proximité temporelle (règles sensori-motrices)
de cohérence structurale et / ou de complétude sémantique
d'isotopie sémantique
fonction du contexte d'interaction
fonction des performances de l'usager [Valot et al., 91]
de logique actionnelle ou intentionnelle [Cohen, 78, 79], [Searle, 83]
etc.

Il est clair que le rôle du module de fusion est de rendre l'interprétation (a) aussi indépendante que possible des contextes dans un premier temps et (b) de permettre une résolution progressive des références pour lever les ambiguïtés dans un deuxième temps. Accessoirement un tel niveau de fusion permet également d'ajouter de nouveaux modes sans avoir à modifier le contrôleur de dialogue en profondeur.

Ces deux contraintes nous conduisent alors à proposer une *fusion progressive* des informations partant des niveaux morphosyntaxiques pour aboutir au niveau sémantique selon le schéma suivant (fig. 4) :

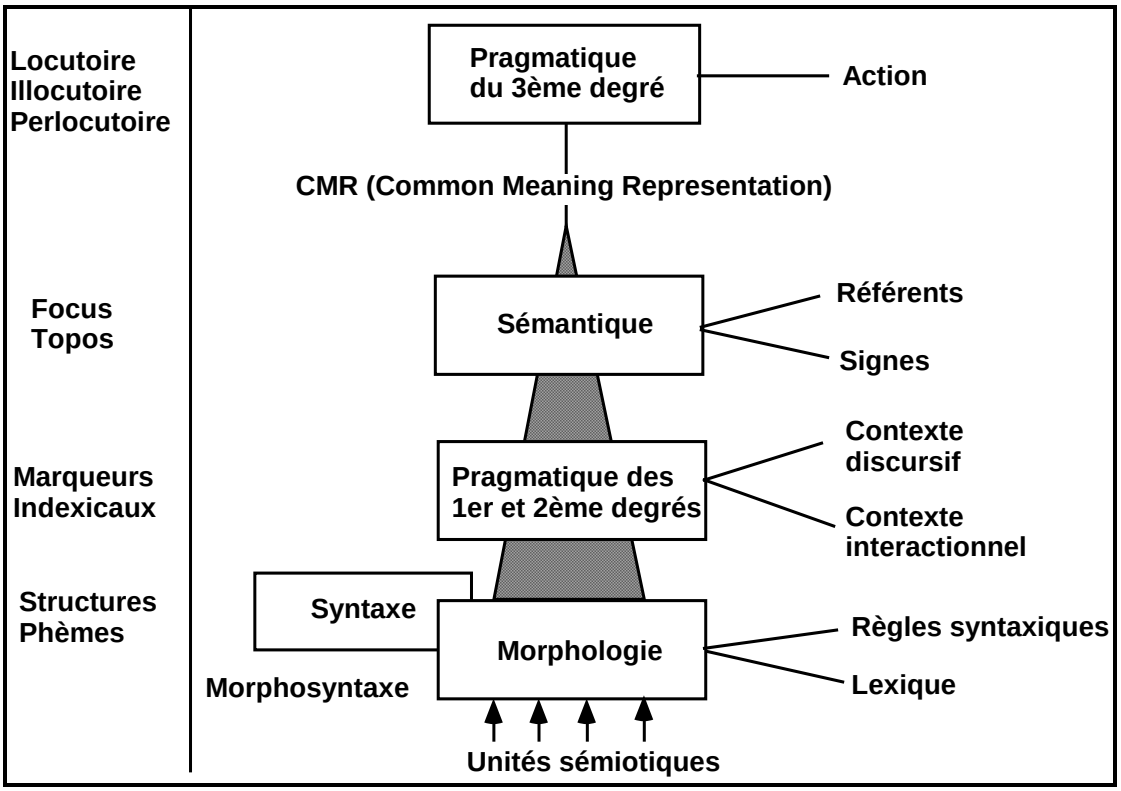


Fig. 4 : Les niveaux de fusion

Dans ce schéma la fusion s'opère à partir des unités collectées dans l'épaisseur du présent et fournit des structures de représentation abstraites (CMR = common meaning representation) débarrassées des composantes modales. Ces structures sont communiquées au contrôleur de dialogue. Détaillons chaque étape de la fusion.

• Analyse morphosyntaxique modale

Une analyse morphosyntaxique de chaque acte modal est faite sur l'épaisseur du présent. On obtient pour chaque mode une représentation adaptée qui décrit la structure des constituants et la structure fonctionnelle.

• Analyse pragmatique des 1er et 2ème degrés

A ce niveau une analyse des indexicaux et des marqueurs pragmatiques par liage intermodal est opérée. Elle permet de relier les éléments référentiels libres d'un mode aux éléments référents des autres modes et de lier les actes entre eux.

• Raisonnement sémantique (spatio-temporel)

Ce raisonnement aboutit à la construction d'une CMR (Common Meaning Representation) par instanciation de schémas (d'action et d'objet). Ces mécanismes ressortissent de mécanismes complexes d'interprétation sémantique du langage naturel [Sabah, 88]. Ils mettent en œuvre des bases de connaissance des actions et des objets ainsi que des règles d'inférence pour instancier ces schémas sur la situation courante. Leur degré de généralité font leur relative indépendance des domaines d'applications.

5. Syntaxe multimodale des énoncés : le côté système

Avec le problème de la syntaxe multimodale, vient le problème de la fusion des modalités : à quel niveau d'abstraction faire coopérer au mieux les informations issues des différents canaux de communication ? Si des approches préconisent la fusion tardive pour gérer les ambiguïtés ou dater les événements, nous avons pour notre part choisi d'effectuer la fusion extrêmement tôt dans le processus d'interprétation.

5.1. Critères d'intégration

Un critère d'interaction définit les conditions pour fusionner les informations provenant de plusieurs modalités. Nous présentons ici ceux qui ont été identifiés lors des travaux de la communauté des Interfaces Homme-Machine [IHM92] :

- La proximité temporelle: sert à mettre en correspondance des événements issus de modalités différentes mais produits en des instants très proches.
- La complémentarité logique (ou structurelle) des événements permet dans certains cas de fusionner, au sein d'une même commande, des événements distants temporellement.
- La complétude d'une structure de données d'intégration peut constituer une condition de passage entre niveaux d'abstraction.
- Les contextes (historique et modèle de la tâche) de dialogue et l'historique d'interaction interviennent dans la résolution des coréférences, des anaphores, des ellipses et des déictiques.
- L'incompatibilité des modalités épargne au processus des tentatives d'intégration de modalités ne pouvant être utilisées simultanément.

Beaucoup de travaux ont porté sur le « temps ». Le temps revêt une grande importance dans les interfaces multimodales, car il devient lui-même porteur d'information, et influe sur l'interprétation des énoncés. Les fig. 9 et 10 montrent qu'à une même séquence d'actions de l'utilisateur, peuvent correspondre deux interprétations différentes, selon la distribution temporelle précise des événements correspondants et en particulier la proximité temporelle de ceux-ci.

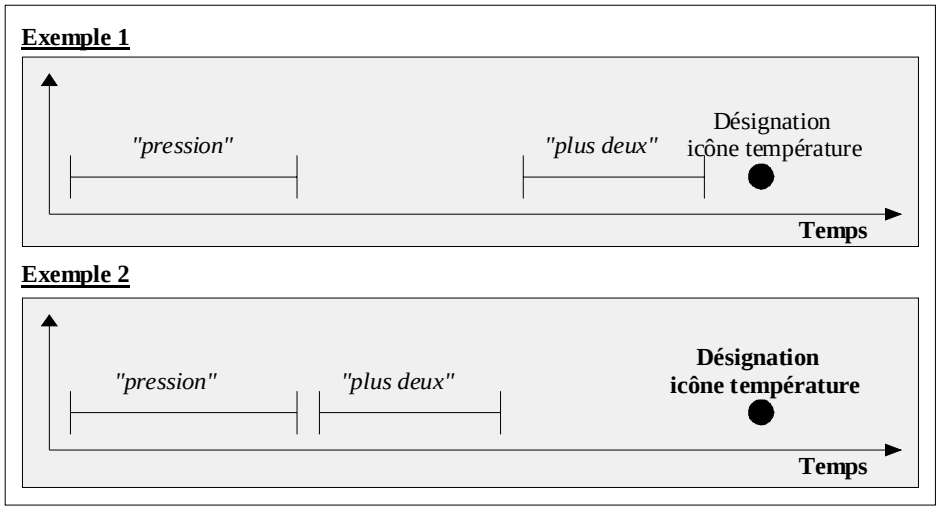


Figure 9. *Importance du temps dans les interfaces multimodales*

Considérons le contexte d'application d'une usine chimique. Dans l'exemple 9 de la fig. **Erreur ! Signet non défini.**, l'utilisateur demande au système de lui communiquer la valeur de la pression, en prononçant le mot "*pression*". La valeur de la pression est alors communiquée à travers le synthétiseur de parole. Puis l'utilisateur décide d'augmenter la température. Disposant sur son écran tactile, d'une icône température (sous forme d'un thermomètre par exemple), il désigne cette icône tout en prononçant les mots "*plus deux*". Le système augmente alors la valeur de la température de 2 unités. Dans l'exemple 2, l'utilisateur prononce d'abord les mots "*pression plus deux*", ce qui a pour effet d'augmenter la pression de 2 unités, puis il désigne l'icône de température. Le système lui communique alors la valeur de la température par l'intermédiaire de la synthèse de parole. Finalement dans l'exemple 1 la température a été augmentée de 2 unités alors que dans l'exemple 2 c'est la pression qui l'a été bien que la séquence des événements soit la même dans les deux cas.

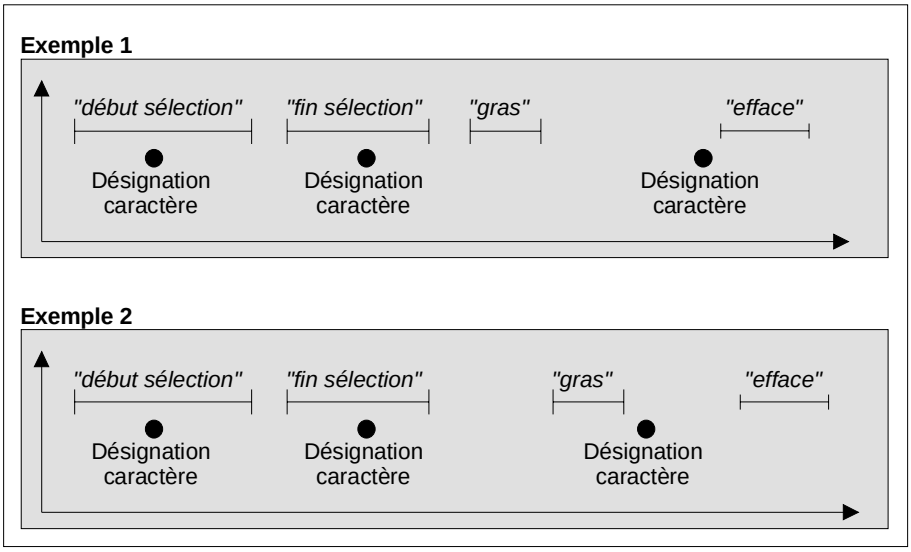


Figure 10. *Importance du temps dans les interfaces multimodales*

Un autre cas illustrant l'importance du facteur temporel et que nous avons rencontré dans MEDITOR [4] est donné dans la fig. 10. Dans le premier exemple l'utilisateur désigne un premier caractère en prononçant la phrase "*début sélection*". Puis il désigne un second caractère en prononçant la phrase "*fin sélection*". Le texte compris entre les deux caractères est alors sélectionné. Il prononce ensuite le mot "*gras*" ce qui a pour effet d'affecter l'attribut gras à la sélection courante. Il désigne ensuite un autre caractère et prononce le mot "*efface*". Seul le caractère désigné est alors effacé. Dans le second exemple, la troisième désignation est effectuée juste après la prononciation du mot gras. Cette proximité temporelle permet à l'utilisateur d'indiquer au système que l'attribut gras doit être affecté au caractère qu'il vient juste de désigner et non à la sélection courante (qui reste toujours valide). Le mot "*efface*" n'étant accompagné d'aucune désignation, il est par conséquent appliqué à la sélection courante. Finalement dans l'exemple 1 la sélection est passée en gras et le caractère a été effacé, alors que dans l'exemple 2 c'est l'inverse qui se produit bien que la séquence des événements soit exactement la même dans les deux cas.

On voit à travers ces exemples que la séquence seule ne suffit pas à interpréter correctement les énoncés multimodaux. Il est nécessaire de connaître la distribution temporelle précise des

informations afin de pouvoir détecter les proximités temporelles entre les événements. Il est par conséquent, indispensable que ces événements soient caractérisés par leurs dates de début et de fin de production. Ceci permet de les classer selon leur ordre chronologique réel et de mesurer les distances temporelles entre eux. Ce type de distance constitue un des critère de fusion des informations. Malheureusement, de nombreux systèmes d'exploitation ne permettent pas d'obtenir une datation précise des événements. Il est alors souvent nécessaire de les contourner, et d'effectuer soi-même une datation approximative à un bas niveau de programmation.

La proximité temporelle

Pour définir concrètement la notion de proximité temporelle, il faut étudier les différents cas de succession de deux messages dans le temps. Allen [5] en a proposé 13 (fig. 11).

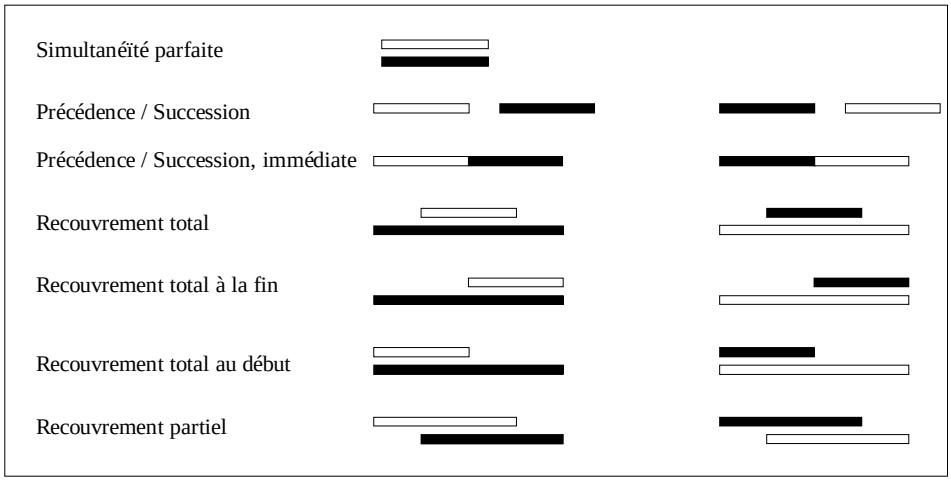


Figure 11. *Les treize relations temporelles d'Allen*

Ces relations nous semblent importantes dans le cas de la multimodalité en sortie ou dans les applications multimédia [6]. Elles permettent de spécifier précisément la manière dont les informations de sortie doivent être synchronisées dans le temps. Cependant, en entrée, nous pensons qu'il n'est pas nécessaire d'en distinguer autant. Les cas que nous avons distingués dans nos réalisations et qui sont présentés dans la fig. 12 nous ont été suffisants.

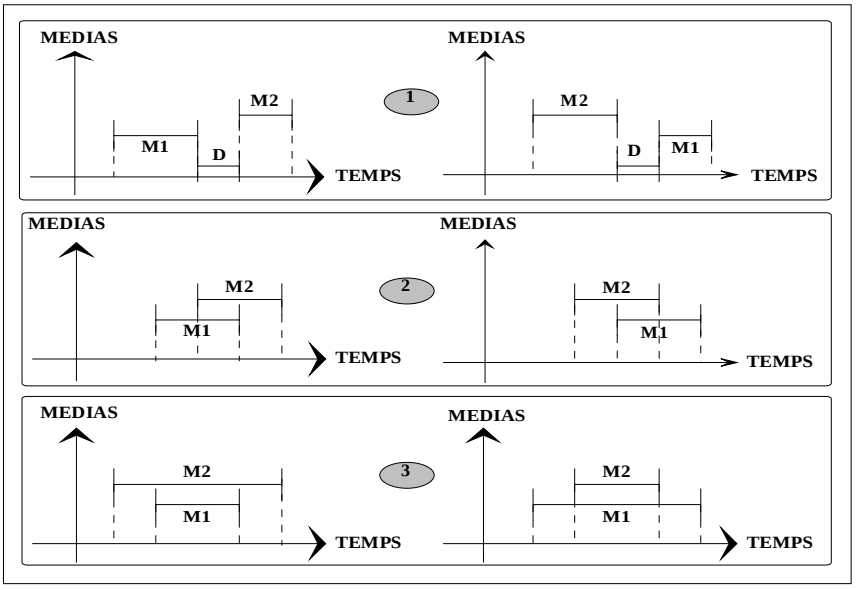


Figure 12. *Succession temporelle de deux messages*

Dans les cas 2 (recouvrement partiel) et 3 (recouvrement total), il est logique de dire que les deux messages sont temporellement proches. Dans le cas 1 (pas d'intersection), la distance temporelle séparant la date de fin de production du premier message et la date de début de production du second est mesurée puis comparée à un seuil déterminé expérimentalement, ou fixé selon les préférences de l'utilisateur.

Temps de réponse des médias d'interaction

Pour pouvoir interpréter correctement les énoncés de l'utilisateur, il est nécessaire de traiter les informations selon leur ordre chronologique réel[1]. Or la différence entre les temps de réponse des différents médias peut être très importante. Ceci implique que le système reçoit en général un flot d'informations dans un ordre qui ne correspond pas au véritable ordre produit par l'utilisateur (fig. 13). Ceci peut conduire à une interprétation erronée des énoncés.

[1]Même pour les être humains, il peut être difficile de comprendre le sens d'une phrase dont l'ordre des mots a été modifié.

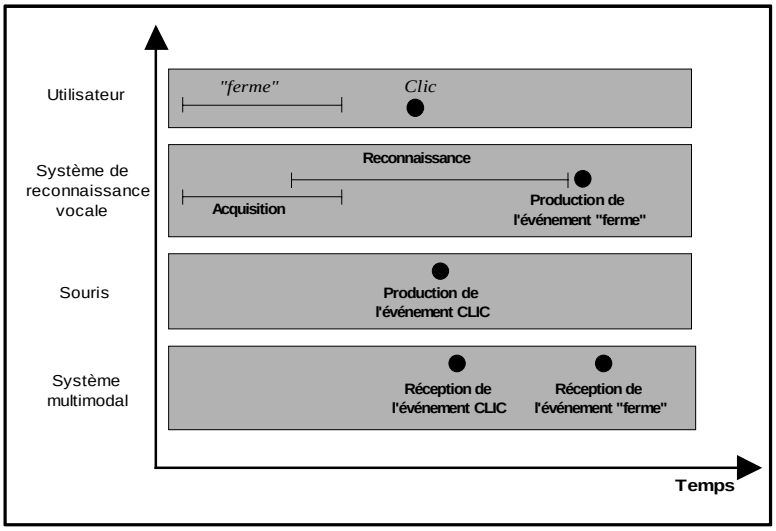


Figure 13. *Problème du temps de réponse des médias*

Un message peut donner lieu à différentes interprétations selon qu'il ait été produit de manière isolée (dans le temps) ou au contraire, en simultanéité avec d'autres messages. Par exemple, la désignation d'une fenêtre peut signifier, lorsqu'elle n'est accompagnée d'aucun ordre vocal, "*mettre en avant-plan la fenêtre désignée*". Par contre, si cette désignation est accompagnée de l'ordre vocal "*ferme*", l'interprétation sera différente. Or, si l'utilisateur prononce le mot "*ferme*" et désigne tout de suite après une fenêtre, on peut alors constater que l'événement correspondant au clic est produit avant l'événement correspondant au mot prononcé, car le système de reconnaissance de parole met beaucoup plus de temps à reconnaître un mot que le driver souris à détecter les coordonnées de pointage. La solution à ce problème consiste tout d'abord à dater les messages (date de début et date de fin) et à maintenir une file des événements triée par ordre chronologique. Ensuite, il convient de ne traiter un événement, qu'après avoir interrogé tous les périphériques pour s'assurer qu'aucun autre événement n'est en cours de production. De cette façon on peut être sûr que le prochain événement qui sera produit aura une date de début de production postérieure à celle de l'événement en cours de traitement. Concernant les systèmes de reconnaissance vocale, tester si un message est en cours de production, signifie tester si l'utilisateur est en train de parler ou si la reconnaissance d'un mot est en cours. Malheureusement cette possibilité n'est pas toujours offerte par les systèmes de reconnaissance.

Coréférences actives et coréférences passives

Nos travaux nous ont amenés à distinguer deux types de coréférence :

1. *les coréférences actives* : correspondent à la production de deux informations à travers deux modalités, telles que l'interprétation et la compréhension complète et sans ambiguïté d'une des informations ne peuvent se faire sans l'autre. Par exemple, l'utilisateur prononce le mot "*ferme*" et clique en même temps sur la barre de titre d'une fenêtre.

2. *les coréférences passives* : correspondent à la production d'une information à travers une modalité, telle que l'interprétation et la compréhension complète et sans ambiguïté de cette information ne peuvent se faire sans connaissance de l'état d'une autre modalité. Par exemple,

l'utilisateur prononce le mot "*ferme*". La fenêtre pointée du regard est alors fermée. Le problème posé par les coréférences passives concerne la sauvegarde des états des périphériques. Pour illustrer ce problème considérons l'exemple suivant dans lequel l'utilisateur dispose, d'un oculomètre et d'un système de reconnaissance vocale. Pour fermer une fenêtre, il prononce "*ferme*" et pointe simultanément du regard la fenêtre désirée (fig. 14).

3.

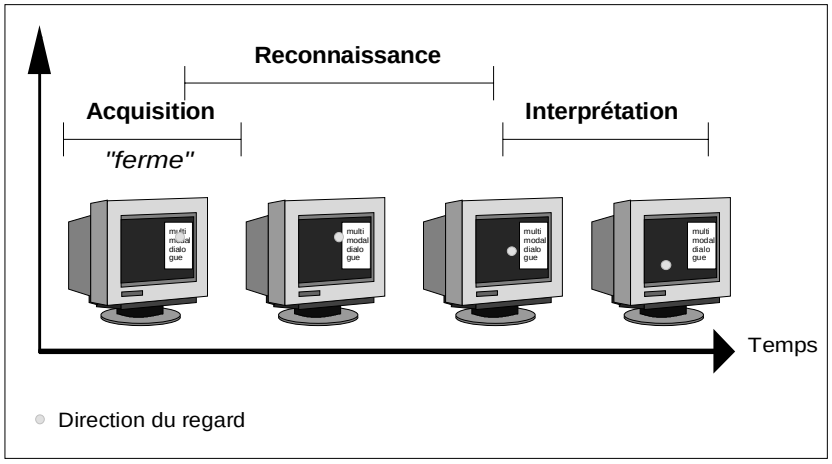


Figure 14. *Problème des coréférences passives*

L'analyse de cette manipulation du point de vue du système, indique que le système de reconnaissance vocale met un certain temps à acquérir et à reconnaître le mot prononcé, et le système multimodal un certain temps à l'interpréter. Aussi courts que soient ces temps, il est très probable, une fois la commande interprétée, que la direction du regard ait changé. Il faut donc retrouver la direction du regard à un instant passé, compris entre l'instant de début de prononciation du mot "*ferme*" et l'instant de fin de prononciation. L'état de oculomètre à cet instant passé permettra de retrouver la direction du regard. La solution à ce problème consiste donc à associer à chaque périphérique à changement d'état fréquent (oculomètre, souris,...) un historique permettant le stockage des états récents ainsi que leurs dates.

Recommandations aux constructeurs de périphériques

Nous résumons dans ce paragraphe les principales informations que les constructeurs de périphériques et les concepteurs de systèmes d'exploitation doivent veiller à fournir afin de permettre une intégration coopérative et une exploitation multimodale des moyens de communication entre l'homme et la machine :

Datation précise des événements : tout structure d'événement doit comporter une date de début de production et une date de fin de production (ou bien une date de début et une durée). Cette datation doit être précise au $1/10^{ème}$ de seconde près.

Historique des événements : pour les périphériques pouvant changer d'état rapidement (oculomètre, souris, stylet, etc.) il est recommandé de disposer d'un historique des états antérieurs d'une durée de 3 ou 4 secondes et à un intervalle minimal de $1/10^{ème}$ de seconde.

Etat des périphériques : il est également nécessaire de disposer d'une information permettant de connaître les divers états possible du périphérique. Par exemple pour un système de reconnaissance de parole, on peut énumérer les états suivants :

- en attente
- en acquisition (l'utilisateur commence à parler)
- en acquisition et reconnaissance (le système commence la reconnaissance avant même que l'utilisateur ait fini de parler)
- en reconnaissance (l'utilisateur à fini de parler mais la reconnaissance n'est pas encore terminée)

Ces états permettront de savoir si un événement E est en cours de production sur un périphérique donné P, auquel cas une éventuelle interprétation d'un autre événement E' issu d'un autre périphérique P' plus rapide que P pourrait être mise en attente jusqu'à délivrance de l'événement E. L'événement E peut en effet, influer sur l'interprétation de l'événement E'.

5.2. Stratégies d'intégration

La stratégie d'intégration peut être précoce ou différée par rapport à la question sémantique. Elle peut aussi être progressive et s'effectuer tout au long des différents niveaux d'abstraction fournis par l'architecture choisie. Dans la liste qui suit, nous donnons diverses stratégies et leurs argumentations :

- Le modèle du creuset, présenté par Laurence Nigay [Nigay, 94] pour l'application MATIS, adopte une stratégie de fusion précoce selon les critères temps, complémentarité et contexte du dialogue. Ce choix implique de défaire parfois certaines fusions mais reste efficace dans le cas général.
- L'intégration à base de règles, développée dans LIMSI-DRAW par Yacine Bellik et Daniel Teil [Bellik, 95], propose une stratégie retardée. Séparée en deux fusions menées en parallèle (fusion locale et fusion globale) la production de l'énoncé ne sera faite qu'au niveau du contrôleur de dialogue. Les critères utilisés diffèrent en fonction de la fusion utilisée. On peut citer la complémentarité logique, la compatibilité des types et la proximité temporelle. Tous les événements arrivant au contrôleur de dialogue sont alors typés, datés et ont une forme commune.
- Le modèle conceptuel de Jean-Claude Martin et Daniel Béroule [Martin, 95] est le seul à proposer une intégration distribuée sur les niveaux d'abstraction consécutifs suivant un critère temporel. La fusion se fait au meilleur moment après l'activation d'un réseau connexionniste.

Nous avons déterminé une stratégie d'intégration qui n'est pas guidée par les modalités à fusionner, mais par les éléments à combiner pour créer une commande. En effet l'interaction multimodale est souvent de nature actionnelle (plus qu'informationnelle). Il est donc intéressant de définir une *logique de l'action* sur laquelle l'utilisateur interagit avec la machine. Nous présentons donc le modèle VA : Verbe-Actants, où le verbe dénomme le type d'action et où les actants dont les attributs de l'action (qui, quoi, quand, où, comment, etc.). Il s'agit d'un processus de fusion précoce où, dès réceptions des signaux, il y a tentative de combinaison sur des entités dépourvues de type au niveau de l'agent d'interprétation. Les critères d'intégration, même s'ils ne sont pas explicitement recherchés dans l'algorithme afin de gagner du temps, sont la proximité temporelle et la complétude structurelle.

Il n'y a pas dans cette approche de modalité dominante puisque très tôt dans le processus les entités

servant à l'interprétation perdent leurs origines et leurs types. En revanche, s'il y a un élément dominant dans ce modèle, c'est le verbe. D'où le problème d'extraction du verbe. S'il est inexistant au niveau du mode gestuel puisque la désignation d'un bouton ou la reconnaissance d'un geste de commande n'amène pas d'importants problèmes de reconnaissance, il est très présent au niveau du langage naturel.

9. Conclusion

Une interface met en relation les niveaux de structuration des connaissances (signes) de mondes référentiels possibles avec les niveaux d'abstraction pour l'architecture de l'interface. Le passage entre ces niveaux (représentations, concepts, symboles) se fait par un double processus : sur l'axe syntagmatique (combinaison des signes, sur l'axe horizontal du temps) par le « dialogue », sur l'axe paradigmatique (combinaison des signes sur l'axe vertical) par le « contrôle ». L'interaction se manifeste par une relation plus directe sur le système matériel, c'est-à-dire que la combinatoire syntagmatique est à plus courte portée et la profondeur des « mondes » moins grande que dans le cas du dialogue. Notons également qu'une interface met en relation plusieurs milieux, celui de l'homme, celui de la machine et celui dans lequel tous deux sont plongés, leur environnement.

Le concepteur d'interfaces doit prendre en compte l'usager dans ses dimensions cognitive mais ici aussi sensorielle et motrice. Cela donne clairement deux niveaux de traitement : (a) un niveau "bas" pour la gestion des modes et la fusion/fission des informations et (b) un niveau "haut" pour la gestion de l'interaction à travers des couches sophistiquées de dialogue.

Nous n'avons pas examiné dans ce cours tous les aspects de la multimodalité. Avec la conception, l'évaluation est une étape fondamentale dans l'élaboration d'une application en vraie grandeur [Coutaz, 90], [Scapin, 86]. On s'aperçoit à ce niveau de l'importance des erreurs de compréhension et du problème de leur réparation [Siroux et al., 89]. Ces erreurs sont non seulement dues aux faiblesses des modules de reconnaissance mais aussi aux phénomènes d'anticipation motrice/concurrence vs. retard/hésitation, aux conflits inter-modaux, aux inattendus.

Bibliographie

[Austin, 62] AUSTIN J.L., How to do things with words. Oxford U. P., 1962

[Barthet, 88] BARTHET M.F., Logiciels interactifs et ergonomie. Modèles et méthodes de conception. Dunor-Informatique, Bordas, Paris, 1988

[Bastide, 91] BASTIDE R., PALANQUE P., "Modélisation de l'interface d'un logiciel de groupe par Objets Coopératifs", document de travail IHM'91 p 1-10.

[Bisson et al., 92] BISSON P., NOGIER J.F., Interaction homme-machine multimodale : le système

MELODIA. Actes ERGO.IA'92, Biarritz, p. 69-90, 1992

[Bourguet et al., 92] BOURGUET M.L. & CAELEN J., "Interfaces Homme-Machine Multimodales: Gestion des Evénements et Représentation des Informations", ERGO-IA'92 proceedings, Biarritz, 1992.

[Bourguet, 92] BOURGUET M.L., Conception et réalisation d'une interface de dialogue personne-machine multimodale. Thèse INPG, Grenoble, 1992

[Buxton, 93] BUXTON B., HCI and the inadequacies of direct manipulation systems. SIGCHI Bulletin, Vol. 25, n°1, p. 21-22, 1993

{Brandetti, 88] BRANDETTI M., D'ORTA P., FERRETTI M., SCARCI S., 1988, "Experiments on the usage of a voice activated text editor", Proc. Speech '88, 1305-1310.

[Brooks, 88] BROOKS F.P., Grasping reality trough illusion : interactive graphics serving science. 5th Conf. on Comp. and Human Interaction, CHI'88, 1988.

{Caelen 91] CAELEN J., Interaction multimodale dans ICPdraw : expérience et perspectives. Ecole de printemps PRC "communication homme-machine", Ecole Centrale de Lyon, 1991.

[Caelen, 92a] CAELEN J., GARCIN P., WRETO J., REYNIER E., Interaction multimodale autour de l'application ICPdraw. Bulletin de la Communication Parlée n°2, p. 141-151.

[Caelen, 92b] CAELEN J., COURAZ J., Interaction homme-machine multimodale : quelques problèmes. Bulletin de la communication parlée n°2, p. 125-140.

[Caelen-Haumont, 91] Stratégie des locuteurs en réponse à des consignes de lecture d'un texte: analyse des interactions entre modèles syntaxiques, sémantiques, pragmatiques et paramètres prosodiques. Thèse de doctorat d'état, vol. I et II, Aix-en-Provence, 1991

[Cadoz, 92] CADOZ Cl., Le geste canal de communication homme-machine. La communication instrumentale. Actes des Entretiens de Lyon, CNRS, 1992.

[Collectif, 91] IHM'91, groupe de travail interfaces multimodales, Dourdan, déc. 1991

[Collectif, 92] IHM'92, groupe de travail interfaces multimodales, Paris, déc. 1992

[Condom, J.M., 92] CONDOM J.M., Un système de dialogue multimodal pour la communication avec un robot manipulateur. Thèse Université P. Sabatier, Toulouse 1992.

[Coutaz et al., 90] COUTAZ J. et CAELEN J., PRC communication homme-machine : Opération de Recherche Concertée interface homme-machine multimodale. Juin 1990.

[Coutaz, 87] COUTAZ J., "PAC: an Implementation Model for Dialog Design", Proceedings of the Interact'87 conference, Stuttgart, H-J. Bullinger, B. Shackel ed., North Holland, september 1987, pp. 431-436.

[Coutaz, 90] COUTAZ J., Interface homme-ordinateur : conception et réalisation. Dunod éd., Paris, 1990.

[Coutaz, 92] COUTAZ J., "Multimedia and Multimodal User Interfaces: A Taxonomy for Software Engineering Research Issues", St Petersburg HCI Workshop, August, 1992.

[Cohen, 78] COHEN Ph.R., On knowing what to say : Planning speech acts. Ph.D. Thesis, Technical Report n°118, Department of Computer Science, University of Toronto, January 1978.

[Cohen et al., 79] COHEN Ph.R. et PERRAULT C.R., Elements of a Plan-Based Theory of Speech Acts. Cognitive Science 3, pp. 177-212, 1979.

[Decouchant et al., 88] D. DECOUCHANT, A.DUDA, A.FREYSSINET, M.RIVEILL, X.ROUSSET de PINA, R.SCIOVILLE, G.VANDOME, "GUIDE: an implementation of the Comandos object-oriented architecture on Unix", Proceedings of EUUG Autumn Conference (Lisbon), p 181-193, October 1988.

[Falzon, 90] FALZON P., Ergonomie Cognitive du Dialogue. PUG, Grenoble, 1990

[Faure, 93] FAURE C., Communication écrite, concepts et perspectives. Journée du GDR-PRC “Communication Homme-Machine”, Montpellier, à paraitre, 1993

[Hécan et al. 75] HECAN H., JEANNEROD M., Du contrôle moteur à l’organisation du geste. Masson éd., Paris, 1975.

[Hutchins, 85] HUTCHINS E.L., HOLLA J.D., NORMAN D.A., Direct Manipulation Interfaces. HCI, Lawrence Erlbaum Ass. Publ., 1(4), 1985, p. 311-339.

[Gaiffe et al., 91] GAIFFE B., PIERREL J.M., ROMARY L., Reference in amultimodal dialogue : towards a unified processing. EUROSPEECH’91, 2nd Euopean Conference on Speech Communication and Technology, Genova, Italy, 1991

[Gourdol, 90] GOURDOL A., Voice Paint, rapport de DEA, Grenoble, 1991

[Grice, 75] GRICE H.P., Logic and conversation. in Syntax and Semantic, 3: Speech Acts, P. Cole and J. L. Morgan (Eds), New York Academic Press, pp. 41-58, 1975.

[Fillmore, C.J.] FILLMORE C.J., The Case For Case. Bach E. and Harms R. eds, “Universals in Linguistic Theory”, Holt, Rinehart and Wiston, pp 1-90, New York, 1968.

[Morel, 88] MOREL M.A., Analyse linguistique d’un corpus de dialogues homme-machine. Publications de la Sorbonne Nouvelle, Tomes I et II, Paris , 1988

[Morel, 89] MOREL M.A., Analyse linguistique d'un corpus, Deuxième corpus: Centre d'Information et d'orientation de l'université de Paris V. Paris: Publications de la Sorbonne Nouvelle, 331 p., 1989.

[Pankoke, 89] PANKOKEe-BABATZ U., "Computer based Group Communication, the AMIGO Activity Model", Ellis Horwood, 1989.

[Reynier, 90] REYNIER E., Analyseurs linguistiques pour la compréhension de la parole. Thèse INPG, Grenoble, 1990

[Rubine, 91] RUBINE D., “The automatic recognition of gesture”, PhD thesis, School of computer Science, Carnegie Mellon University, CMU-CS-91-202, 1991.

[Scapin, 86] SCAPIN D.L., “Guide ergonomique de conception des interfaces homme-machine”, Rapport Technique INRIA no 77, Octobre 1986

[Taylor et al., 89] TAYLOR M.M., NEEL F., BOUHUIS D.G., The Structure of Multimodal Dialogue. Elsevier Science Publishers B.V., North-Holland, 1989

[Sabah, 88] SABAH G., L'intelligence artificielle et le langage. 2 tomes. Hermès ed., 1988 et 1989.

[Searle, 69] SEARLE J.R., Speech Acts. Cambridge U. P., 1969

[Searle, 83] SEARLE J.R., Intentionality. Cambridge U. P., 1983.

[Siroux et al., 89] SIROUX J., GILLOUX M., GUYOMARD M., SORIN C., Le dialogue homme-machine en langue naturelle : un défi ? Annales des télécommunications, 44, n°1-2, 1989.

[Stefik et al.,87] STEFIK M., BOBROW D., FOSTER S., TATAR D., "WYSIWIS: Early experiences with multi-user interfaces" ACM trans. office information system, Vol.5, n°2, April 1987, p 147-167.

[Turk, 91] TURK M. and PENTLAND A., “Eigenfaces for recognition”, Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.

[Valot et al., 91] VALOT C., AMALBERTI R., Description et analyse de l’activité de l’opérateur. Ecole IHM-M, Ecole Centrale, Lyon avril 1991

[Vernant, 92] VERNANT D., Modèles projectifs et structure actionnelle du dialogue. in Recherches sur la philosophie et le langage, Du Dialogue, Vrin éd., 1992.

[Wilson, 91] WILSON M.D., An architecture for multimodal dialogue, Workshop ESCA, Venaco, 1991

Annexe

Un exemple : dialogue dans l'éditeur ICPdraw

ICPdraw est une application de dessin (type MacDraw) dans lequel la communication homme-machine est multimodale. L'utilisateur dispose d'une palette d'outils graphiques et de menus de fonctions. Il peut aussi activer ces fonctions par la parole ou l'écriture.

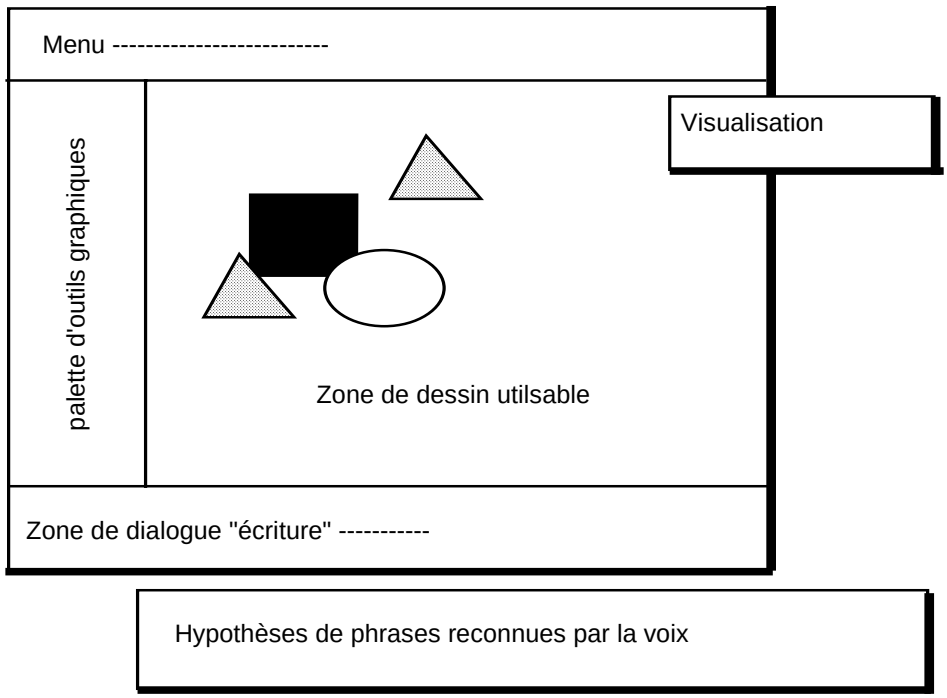


Fig. 5 : Exemple d'écran pour l'application ICPdraw. Il est composé de trois fenêtres, la première, découpée en zones et définissant l'espace de travail graphique, clavier-souris, la deuxième pour visualiser les événements multimodaux et pour indiquer à l'utilisateur le moment où il peut intervenir, la troisième pour visualiser les résultats de la compréhension de la parole.

Le langage de manipulation

Le langage oral de manipulation (dessin, déplacement, coloriage, etc.) des objets géométriques de ICPdraw est défini de la manière suivante :

la structure logique de la commande est :

Verbe(<arg_1><arg_2>...<arg_n>)

Verbe représente une tâche élémentaire ou une succession de tâches à effectuer. C'est très souvent le verbe de la phrase

arg_i sont des arguments de la fonction Verbe. Ils sont de type GN ou GP, le Nom du GN est en général un objet de l'application et Adjectif un attribut de cet objet lorsque Nom et Adj sont dans le même GN.

Les mots-outils sont facultatifs dans un tel langage

Ex: "dessine un cercle de couleur noire": (un=quelconque) (1)
"détruis le cercle": (le=celui dont il vient d'être question)
"dessine un cercle noir": autre forme de (1)
"dessine cercle noir": forme abrégée de (1) sans article

On ne s'intéresse pas dans la suite au module "Parole" On suppose que ce module dispose d'analyseurs linguistiques capables de fournir la structure des constituants (c-structure) et la structure fonctionnelle (f-structure) de la commande énoncée par la voix. Par exemple pour la phrase "dessine un cercle de couleur noire" cela donne (Fig. 6) :

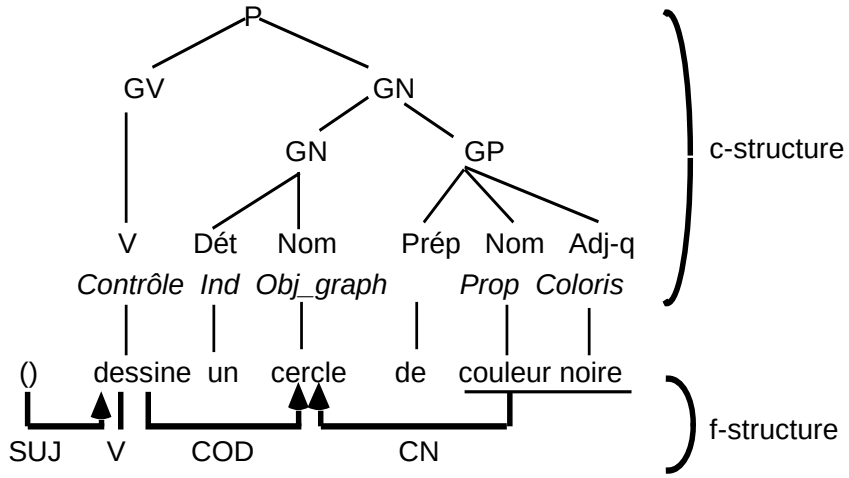


Fig. 6 : En supposant la phrase correctement comprise les analyseurs linguistiques du module Parole délivrent la c-structure (ou arbre des catégories syntaxico-sémantiques) et la f-structure (relations fonctionnelles comme sujet=SUJ, COD=complément d'objet direct, CN=complément de nom) de la phrase P.

Cette analyse ne suffit pas en général, il faut encore interpréter cette commande dans le contexte de l'application. Pour cela on utilise généralement des grammaires de cas pour retrouver la forme logique V(). Ces grammaires de cas sont mises en œuvre à travers des bases de connaissance et un analyseur. L'analyseur gère lui-même son propre historique pour résoudre les problèmes d'ellipse et d'anaphore au cours du dialogue.

Le contrôleur du dialogue peut vérifier si une action est exécutable (toutes les conditions sont requises, tous les arguments sont valués) ou prédire une action. Ce contrôleur se présente donc comme un ATN ou un GPS, ou un planificateur.

Pour l'exemple ci-dessus il faudra obtenir le schéma d'action :

```
dessiner : sorte_de 'dessin'
      a-attributs
            objet (quoi) = cercle.5
            destinataire (à qui) = ?
            agent (qui) = 'système'
```

manière (comment) = ?
temps (quand) = 'immédiat'
cause (pourquoi) = ?
lieu (où) = ?
quantité (combien) = 1
but (pour) = ?
condition (si) ='néant'
concession (malgré) = 'néant'
restriction (sauf) = 'néant'
destination (vers) = ?

et instancier le nouvel objet :

cercle.5 : sorte_de 'obj_graph'
a-forme = cercle
a-taille = ?
a-couleur_fond = noir
a-contour = ?

De manière générale l'analyseur devra disposer d'une base d'objets et d'une base de tâches ainsi que de mécanismes de remplissage des slots. Ce mécanisme ne sera pas détaillé ici.

(a) la base d'objets

Définition des objets :

Objet: sorte_de 'classe'
a-attributs (caractéristiques et contraintes)
m-méthodes (liste des méthodes attachées à l'objet)
s-liens sémantiques
c-contraintes (ou restrictions sur attributs des classes pères)

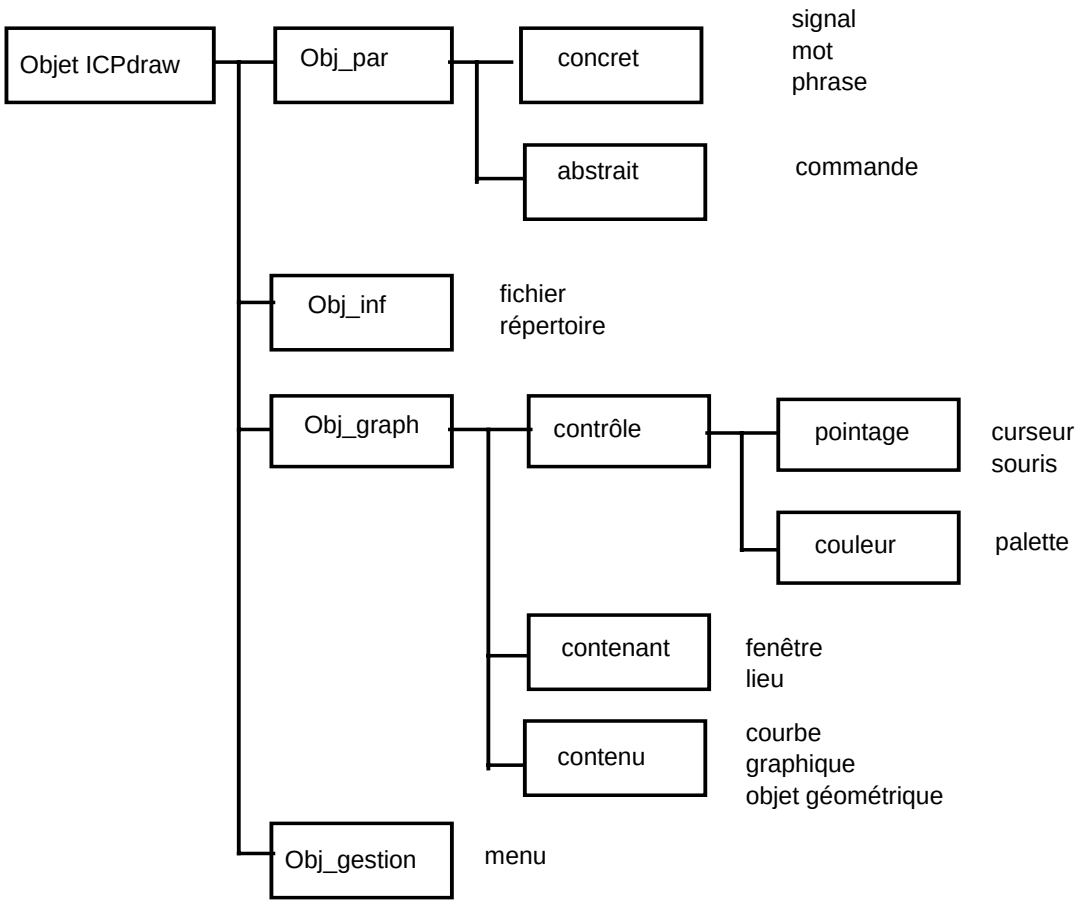


Fig. 7 : Structuration de quelques objets d'ICPdraw

```
Obj_inf: sorte_de 'obj ICPdraw'
        a-nom: char(32)
        a-date: jour / mois / année
        a-propriétaire: char(20)
        a-taille: entier [octets]
        a-privilège {public, privé}

fichier: sorte_de 'obj_inf'
        m-{ouvrir, fermer, sauver, dupliquer, renommer, imprimer, lister }
        s-CONTIENT('données$type')
        s-EST_CONTENU('répertoire')
        c-taille > 0

répertoire: sorte_de 'obj_inf'
        a-niveau: entier
        m-{ouvrir, fermer, sauver, examiner, renommer, imprimer, lister}
        s-CONTIENT('fichier')

parole_compressée: sorte_de 'fichier'
        a-type: {amplitude, fréquence, indice}
        a-entête:
                titre: char(80)
                Nb_éch: entier > 0
                Nb_bits: entier (1,32)
```

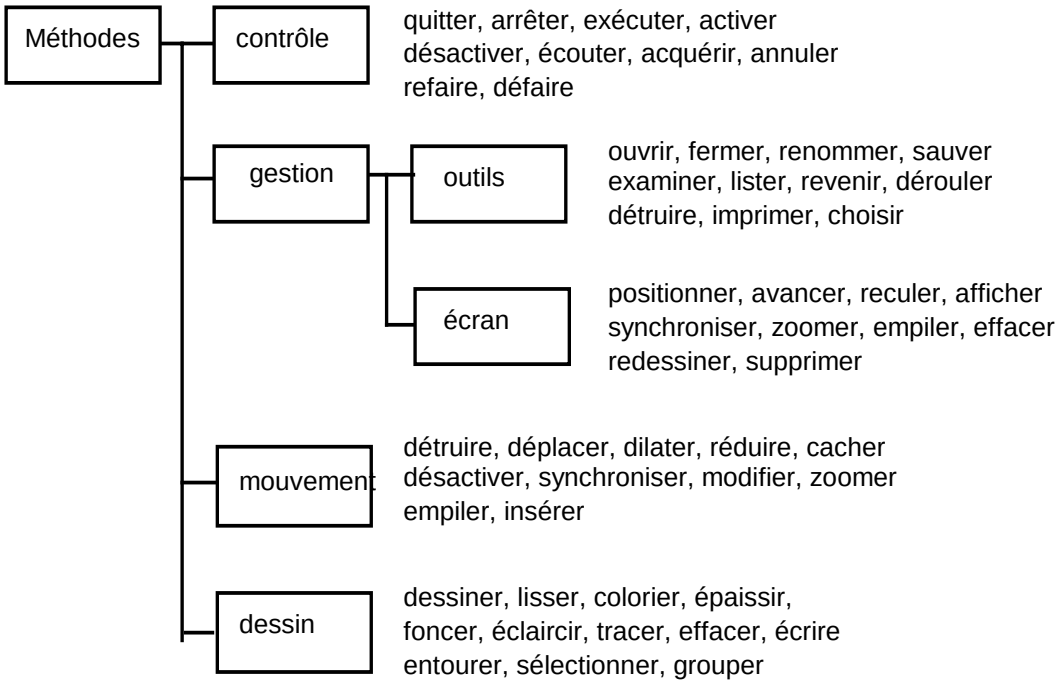
```
        Fe: entier [Hz]
    a-taille_enreg: entier
    m-{écouter, sauver}
    s-SYNCHRONE($type, 'temps')
```

etc.

(b) la base des tâches

Les tâches peuvent être vues comme des procédures activables par le verbe de la commande et dont les actants sont les arguments de la commande : c'est typiquement une formulation casuelle. Dans cette perspective la syntaxe des schémas des tâches est la suivante :

```
_____________________________________________________
méthode: sorte_de 'classe'
    a-attributs
        objet (quoi)
        bénéficiaire (à qui)
        agent (qui fait)
        patient (qui subit)
        manière (comment)
        temps (quand)
        cause (pourquoi)
        lieu (où)
        quantité (combien)
        but (pour)
        condition (si)
        concession (malgré)
        restriction (sauf)
        destination (vers), etc.
    s-liens sémantiques
    c-contraintes
_____________________________________________________
```




```
action_contrôle: sorte_de 'méthode'

quitter: sorte_de 'action_contrôle'
        quoi: 'application'
        quand: 'immédiat'
        vers: 'système'

écouter: sorte_de 'action_contrôle'
        quoi: 'signal'
        comment: 'mode_interruption'
        quand: APRES('sélection')
        où: 'fenêtre_signal'
        combien: $Nb
        si: Durée_sélectée > 0 ET etc.
        vers: codage_analogique

ouvrir: sorte_de 'action_gestion_outils'
        quoi: $COD OU Historique
        quand: 'immédiat'
        combien: 1
        si: EXISTE($COD(a-nom))
```

etc.