ATELIER IHM'92
ENST

30 Novembre - 2 Décembre 1992

INTERFACES MULTIMODALES
et
ARCHITECTURE LOGICIELLE

Coordinateur : Joëlle Coutaz (LGI-IMAG).

Comité de lecture : Thomas Baudel (LRI), Noëlle Carbonell (INRIA-CRIN), Jean-Claude Martin (LIMSI), Nadine Vigouroux (IRIT).

Rédacteurs : F. Azémard (IRIT), T. Baudel (LRI), Y. Bellik (LIMSI), M-L. Bourguet (ICP), N. Carbonell (INRIA-CRIN), J. Coutaz (LGI-IMAG), J-C. Martin (LIMSI), L. Nigay (LGI-IMAG), K. Ouadou (ECL), P. Palanque (LIS), D. Teil (LIMSI), N. Vigouroux (IRIT).

Participants : F. Azémard (IRIT), T. Baudel (LRI), Y. Bellik (LIMSI), S-K. Bennacef (LIMSI), M-L. Bourguet (ICP), N. Carbonell (INRIA-CRIN), J. Coutaz (LGI-IMAG), V. Gaildrat (IRIT), J-C. Martin (LIMSI), L. Nigay (LGI-IMAG), K. Ouadou (ECL), P. Palanque (LIS), D. Teil (LIMSI), N. Vigouroux (IRIT).

1. Introduction

Ce document concrétise les résultats de l'atelier Interfaces Multimodales et Architecture Logicielle organisé dans le cadre de IHM'92, quatrièmes journées sur l'Ingénierie des Interfaces Homme-Machine.

Une Interface Homme-Machine Multimodale (IHMM) autorise l'utilisation conjointe des différentes modalités sensori-motrices de la communication humaine tels la vision, la langue naturelle et le geste. Elle va au-delà des interfaces multimédia puisqu'elle est capable d'interpréter les informations échangées sur les différents canaux de communication. La réalisation des IHMM introduit de nouvelles difficultés, et notamment l'intégration de modalités multiples au sein d'une architecture appropriée. Notre groupe de travail s'est fixé comme objectif l'analyse de ce processus d'intégration.

Pour cela, nous adoptons comme fondement le modèle d'architecture Arch que nous présentons au paragraphe 2. Nous constatons que l'intégration des modalités présente un certain nombre de caractéristiques font l'objet du paragraphe 3. Elles sont illustrées au paragraphe 4 au moyen de solutions concrètes implémentées par les participants de l'atelier. En conclusion, on trouvera aux paragraphes 5 et 6 une synthèse sur le processus d'intégration ainsi qu'une liste de problèmes non traités pouvant servir d'agenda de recherche à moyen terme en liaison avec les rapports des autres ateliers de travail [Capdevielle 93], [Duby 93].

2. Le Modèle de référence : Arch [J. Coutaz, LGI-Imag]

2.1. Les éléments du modèle Arch

Comme le montre la figure 1, le modèle Arch [UIMS 92] s’appuie sur les composants conceptuels du modèle Seeheim [Pfaff 85]. On y retrouve les notions de noyau fonctionnel, de contrôleur de dialogue et de présentation.

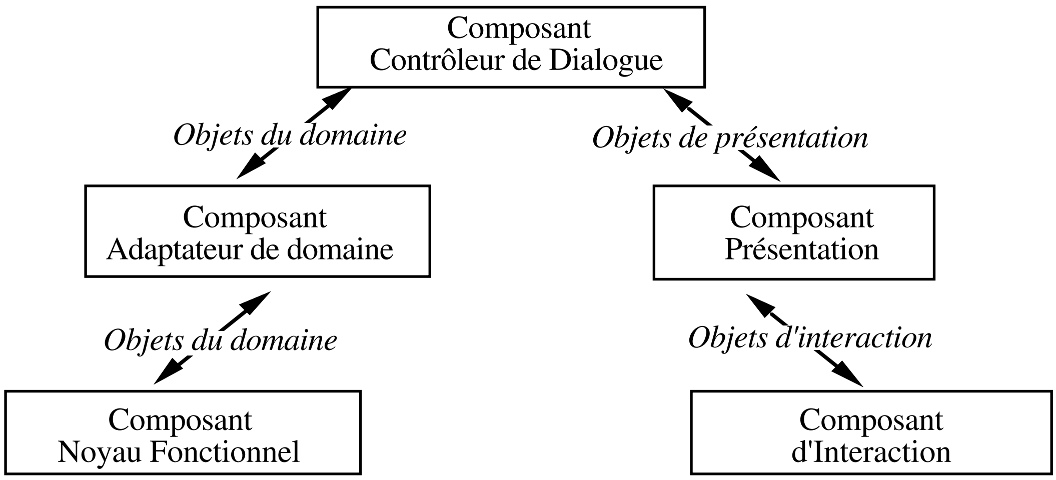


Figure 1 : Les composants du modèle Arch.

Les pieds de l’arche constituent les composants imposés par la réalité : le noyau fonctionnel réalise les concepts du domaine ; le composant d’interaction, en contact direct avec l’utilisateur, est mis en œuvre au moyen des objets d’interaction d’une boîte à outils. En clé de voûte, le contrôleur de dialogue gère l’enchaînement des tâches ainsi que les liens entre les objets regroupés dans les deux composants voisins : l’adaptateur de domaine et la présentation.

L’adaptateur de domaine sert, pour l’essentiel, à ajuster les différences de modélisation des objets conceptuels entre le noyau fonctionnel et le contrôleur de dialogue. Le composant de présentation joue un rôle similaire : il permet au contrôleur de dialogue de s’affranchir du fonctionnement de la boîte à outils du niveau interaction. La présentation peut se voir comme une boîte à outils virtuelle qui implémente des objets de présentation concrétisés en fin de compte par les objets d’interaction d’une boîte à outils réelle.

La figure 1 doit se voir comme une représentation canonique. En réalité, l’importance relative des cinq composants de Arch varie en fonction du cas à traiter. Cette migration des fonctions entre les composants s’exprime métaphoriquement au moyen du métamodèle *slinky*, du nom de ce jouet qui, une fois mis en mouvement, voit sa masse se déplacer dynamiquement. Le choix de la répartition des fonctions entre les composants relève du savoir-faire en ingénierie du logiciel. L’équilibre doit être le résultat d’un dosage analytique des facteurs qualité comme l’efficacité et la portabilité.

Nous appelerons *instance d’Arch*, un exemplaire de modèle Arch produit en appliquant slinky à un système interactif particulier. Nous conviendrons aussi que l’*interface homme-machine* (IHM) d’un système regroupe tous les composants d’une instance d’Arch à l’exception du noyau fonctionnel.

2.2. Utilisation du modèle Arch dans l'atelier

Le modèle Arch et slinky, son compagnon, nous permettent de préciser les notions d'événement et d'énoncé. Ils mettent également en évidence des niveaux privilégiés pour l'intégration des modalités. Ces notions sont à stabiliser pour répondre à un concensus commun.

2.2.1. Evénement

Un événement est l'unité d'information qui transite entre les composants interaction et présentation du modèle Arch. On le caractérise en général par un type, une date et un contenu informationnel (ou valeur).

Le niveau d'abstraction du contenu informationnel dépend de l'instance d'Arch considérée. Par exemple, selon les outils graphiques employés au niveau interaction, l'événement traduit un simple clic souris ou bien véhicule l'identité de l'objet désigné au moyen de la souris ; ou encore, selon le système de reconnaissance de la parole du composant d'interaction, le contenu informationnel contient un mot ou une phrase.

2.2.2. Enoncé

Un énoncé est l'unité d'information qui transite entre le contrôleur de dialogue et l'adaptateur de noyau fonctionnel. Il traduit un changement d'état de l'IHM dont l'effet doit se poursuivre dans le noyau fonctionnel ou, dans le sens inverse, un changement d'état du noyau fonctionnel qui doit être reflété dans l'IHM.

La nature de l'énoncé dépend de l'instance d'Arch. Dans les exemples de réalisation de notre groupe de travail, il modélise généralement une commande complète dont l'effet sémantique doit être réalisé par le noyau fonctionnel. Cependant, nous n'avons pas de réponse claire sur la nature des énoncés émanant du noyau fonctionnel.

2.2.3. Niveaux d'intégration

La fusion des événements en énoncés, et le phénomène inverse, est un processus procédant par étapes. Les composants du modèle Arch définissent des niveaux de référence pour ces étapes.

3. Caractéristiques de l'intégration des modalités [K. Ouadou, ECL & T. Baudel, LRI]

Notre analyse sur l'intégration des modalités s'appuie sur une taxonomie des interfaces multimodales ébauchée lors des journées de travail IHM'91. On trouvera dans [Bourguet 92a, Nigay 92], un affinement de cette proposition initiale. Il ressort de cet effort de classification deux composantes intéressantes pour la mise en œuvre des interfaces multimodales : l'usage temporel (c.-à-d. parallèle ou séquentiel) des dispositifs d'entrée/sortie et l'usage d'une ou de plusieurs modalités mises en jeu dans l'interaction.

Les systèmes synergiques se caractérisent par un usage parallèle de plusieurs modes de communication et une interprétation combinée des différents événements monomodaux émis sur les médias en vue de construire des énoncés multimodaux.

Dans l'atelier, nous nous sommes intéressés à ce processus d'interprétation. Par manque d'expérience, nous n'avons pas étudié le phénomène inverse qui, partant d'un énoncé produit par le noyau fonctionnel, aboutit à une génération multimodale puis une réalisation sur les différents médias disponibles. Nous présentons ci-dessous une liste de caractéristiques propres des systèmes multi-modaux. Celle-ci nous permettra de comparer les différents modèles d'intégration de modalités présentés lors de l'atelier.

3.1. Type de système multimodal

Quatre types de systèmes ont été identifiés : les systèmes de multimodalité alternée, exclusive, synergique ou concurrente. Un système donné peut n'autoriser qu'un seul type de multimodalité. Un autre peut être hybride et permettre, selon les contextes, plusieurs types de multimodalité. Dans ce cas, le modèle d'intégration doit être capable de gérer plusieurs stratégies de fusion et de décider dynamiquement de la stratégie appropriée.

3.2. Existence d'une modalité dominante

L'usage préférentiel d'un mode influe sur l'organisation de l'interaction. On parlera dans ce cas d'une modalité dominante. Les autres modalités disponibles sont utilisées plutôt pour compléter la modalité dominante. Une modalité dominante peut servir de pivot d'intégration, ce qui se traduit à la fois au niveau de la stratégie d'interprétation et du choix des représentations des informations multimodales. On trouvera dans le chapitre 4 des exemples de système. A titre d'exemple, les premiers systèmes multi-modaux [Bolt 80] utilisaient la désignation gestuelle uniquement comme complément à la reconnaissance de la parole. La modalité vocale avait donc un caractère dominant sur les autres modalités prises en compte et supportait le fil directeur de l'interaction. [Baudel 91] et [Baudel 93] proposent d'autres modèles d'interaction dans lesquels au contraire le geste est une modalité dominante, la parole ne servant qu'à compléter et à rendre l'interaction plus directe.

3.3. Stratégie d'intégration

La stratégie d'interprétation se définit par :
• les niveaux d'abstraction de ARCH concernés par le mécanisme d'interprétation (a),
• et l'espace des possibilités de relations intermodales (b).

3.3.1. Niveaux d'abstraction de ARCH (a)

La stratégie d'intégration peut conduire à une intégration soit *progressive* , soit *unique* (dans un seul composant), le long des différents niveaux d'abstraction de Arch. Elle peut être *précoce* ("eager") ou *différée* ("lazy").

Une stragégie précoce a lieu dès la couche d'interaction de Arch. Elle tente de fusionner les événements issus du composant d'interaction et conduit généralement à une fusion progressive tout au long des différents niveaux d'abstraction de Arch. Une stratégie différée a lieu dans le contrôleur de dialogue au niveau d'abstraction le plus haut.

Une stratégie précoce semble offrir les moyens de produire les retours d'information réactifs de la manipulation directe qui ne relèvent pas de la sémantique du domaine. A l'inverse de la stratégie retardée, une stratégie précoce peut amener à défaire des fusions.

3.3.2. Espace des possibilités des relations intermodales (b)

Le processus d'intégration élabore un espace de possibilités de relations intermodales. Cet espace est parcouru soit *en largeur d'abord*, soit en *profondeur d'abord* .

La stratégie, *en largeur d'abord*, identifie les relations possibles entre les modalités à un niveau d'abstraction donné et choisit la meilleure avant de passer au niveau suivant (et ceci de manière répétitive jusqu'à atteindre le noyau fonctionnel).

La stratégie de fusion, *en profondeur d'abord*, exploite la première relation (souvent temporelle) et l'enrichit à travers les niveaux d'abstraction en fonction des relations structurelles et sémantiques. C'est seulement au niveau sémantique que sont validées ou réfutées les interprétations. Les retours arrière dans l'espace des possibilités se font progressivement à travers les niveaux.

3.4. Critères d'intégration

Un critère d'intégration définit une condition pour fusionner ou non des informations. Nous avons identifié les critères suivants :

• *la proximité temporelle* sert à mettre en correspondance des événements issus de modalités différentes mais produits en des instants très proches. Cette proximité temporelle peut se définir par plusieurs paramètres, à savoir : une fenêtre de détection et une limite d'attente d'événements multimodaux.
• *la complémentarité logique* (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. Cette complétude peut servir également à limiter l'attente d'événements.
• *les contextes* (historique et modèle de la tâche) de dialogue et l'historique d'interaction interviennent dans la résolution des coréférences intermodalités, des anaphores, des ellipses et des déictiques.
• *l'incompatibilité des modalités* épargne au processus des tentatives d'intégration de modalités ne pouvant être utilisées conjointement.

3.5. Représentation des Informations

La technique de représentation précise la nature des structures de données employées comme support aux informations fusionnées. Le formalisme de représentation peut être *unique* (quel que soit le niveau d'abstraction) ou *multiple* (adapté au niveau d'abstraction).

3.6. Remarques

Cette liste de concepts n'est pas exhaustive : elle doit être précisée, validée et poursuivie au cours des prochains ateliers. Egalement, l'effort de mise en place d'une terminologie commune doit être continué. C'est sur cette base de concepts relevant d'un consensus que l'atelier a pu aborder et clarifier le processus d'intégration des modalités, exposé par les différents intervenants. C'est ainsi que la présentation des différents systèmes, tous décrits selon une même grille a permis de mettre en évidence des points communs, des divergences et des aspects complémentaires des différentes solutions proposées (Voir Paragraphe 5).

4. Intégration des modalités : exemples de solutions

Les expériences et solutions présentées ici sont le fruit d'études menées par les membres de l'atelier. Chaque approche se situe dans la grille d'analyse de la section précédente et montre comment est réalisée la fusion pour le cas d'école “mets ça ici” et pour un exemple choisi par l'auteur pour mettre en évidence ses spécificités.

4.1. MATIS : un système d'information multimodal supportant des commandes dictées, écrites ou spécifiées par manipulation directe [L. Nigay & J. Coutaz, LGI]

MATIS (Multimodal Airline Travel Information System) est un système d'information sur les transports aériens. Il fournit, en réponse à des requêtes de l'utilisateur, des informations sur les vols entre deux villes. La base de données sous-jacente contient actuellement neuf villes américaines, neuf compagnies aériennes et des informations sur chaque vol telles que le numéro du vol, les repas servis à bord, etc. L'utilisateur exprime des requêtes au système afin de planifier son voyage. Pour cela :

• L'utilisateur peut utiliser la souris et spécifier sa requête par manipulation directe.

• L'utilisateur peut utiliser le clavier et spécifier sa requête en langue naturelle ou en remplissant un formulaire.

• L'utilisateur peut utiliser la parole et dicter sa requête. Par exemple :
<I would like to know the Usair flights from Pittsburgh to Boston>. Le système de reconnaissance utilisé est Sphynx [Lunati 91], un système qui assure une reconnaissance multilocuteurs de la parole continue.

• L'utilisateur peut utiliser les trois modes de communication précédemment cités en les combinant. Par exemple :

<Parole : *I would like to know the Usair flights from this city* >
+ <Sélection avec la souris d'une ville>
+ <Parole : *arriving at*>
+ <Saisie d'un horaire en utilisant le clavier>

MATIS a permis une étude sur l'intégration des modalités (mécanisme de fusion des événements, références multimodales) ainsi que sur l'architecture logicielle des systèmes multimodaux.

4.1.1. Modèle d'architecture logicielle : PAC-Amodeus

Le modèle d'architecture conceptuel appliqué est PAC-Amodeus [Nigay 91]. PAC-Amodeus reprend les composants du modèle ARCH (cf paragraphe 2.1) dont il organise le composant Contrôleur de Dialogue en une hiérarchie d'agents PAC [Coutaz 87]. Cette hiérarchie a deux niveaux. Les feuilles de la hiérarchie sont les agents suivants :

• Les agents *Requête* correspondent aux requêtes en cours de spécification.

• Les agents *Résultat de requête* correspondent aux fenêtres affichant les réponses à une requête.

• L'agent *Retour d'Information* assure l'affichage d'un retour d'information sur tous les actes de parole (analyse syntaxique de la phrase dictée).

• L'agent *Bloc-notes* gère un bloc-notes où l'utilisateur peut noter des informations relatives à son voyage.

• L'agent *Historique* gère l'historique des requêtes émises (cet historique est éditable).

• L'agent *Palette* correspond à la palette d'outils offerts à l'utilisateur pour spécifier une requête, comme par exemple la liste des compagnies aériennes et la liste des villes.

Enfin l'agent *Ciment syntaxique*, racine de la hiérarchie, permet de cimenter les actions de l'utilisateur reçues et traitées par ses agents fils et communique avec le composant Adaptateur de Domaine lors de l'envoi d'une requête.

La partie Contrôle de chaque agent contient un moteur de fusion qui manipule des informations stockées dans un "blackboard". Le paragraphe suivant décrit le moteur de fusion ainsi que la représentation des informations contenues dans le "blackboard".

4.1.2. Stratégie et Critères d'intégration

Comme le montrent les exemples d'énoncés de requêtes décrits en introduction, MATIS est un système multimodal de type synergique. Il supporte cependant la spécification en parallèle de deux requêtes indépendantes, il est donc aussi de type concurrent.
MATIS ne privilégie pas un mode de communication particulier, l'utilisateur peut par exemple ne travailler qu'avec le clavier ou alors utiliser MATIS comme un système uniquement vocal.

Le statégie du moteur de fusion est précoce (eager). Le moteur ne se met jamais en attente d'une information complémentaire. Ceci implique par contre de défaire certaines fusions incorrectes. C'est parfois le cas lorsque les informations ne sont pas traitées dans leur ordre chronologique d'acquisition. (Cet ordre est remis en cause par les différences de temps d'acquisition des événements selon les modalités.)

Les critères d'intégration du moteur de fusion sont le temps, la complémentarité structurelle et le contexte du dialogue. Ces trois critères définissent trois niveaux de fusion : La Micro Fusion, la Macro Fusion et la Fusion Contextuelle. Si les structures logiques de deux événements ne sont pas compatibles, leur fusion est impossible. La complémentarité structurelle est donc une condition nécessaire mais non suffisante à tous les niveaux de fusion. Les trois niveaux de fusion sont présentés dans un ordre décroissant de priorité :

• La Micro Fusion se base sur le temps et combine des événements ayant des intervalles de temps qui s'entrelacent (parallélisme ou pseudo-parallélisme).

• La Macro Fusion se base sur le temps et combine des événements qui appartiennent à une fenêtre temporelle (proximité temporelle).

• La Fusion Contextuelle utilise uniquement le contexte courant de l'interaction. Dans MATIS, ce contexte est réduit à la requête courante.

4.1.3. Représentation des informations

Le moteur de fusion manipule des informations stockées dans un *blackboard*. Une structure de représentation commune unique, appelée Creuset, est utilisée. Cette structure sert à représenter l'information à tous les niveaux d'abstraction : les informations à fusionner et le résultat de la fusion. Ainsi le résultat de la fusion de deux Creusets est un Creuset. Ce choix de représentation unique permet de défusionner facilement, les informations étant toutes exprimées dans le même formalisme. Un Creuset correspond à un ensemble de cases. Aucune sémantique particulière n'est attachée à une case. Dans le cas de MATIS, la case numérotée 1 correspond à la ville de départ. Le moteur de fusion manipule cette case sans connaître la sémantique de l'information stockée. Dans une autre application, comme par exemple ICP-Plan, la case numérotée 1 permettra de stocker un objet graphique (mur, porte par exemple). Ainsi cette structure de représentation, appelée Creuset, est indépendante du domaine de l'application.

Le moteur de fusion, de part ses critères d'intégration et de sa représentation des informations dans des Creusets, est donc indépendant du domaine de l'application.

4.1.4. Exemples

Exemple 1 **:** L'exemple de fusion "mets ça là" est détaillé dans [Nigay 92]. A l'origine, trois Creusets sont créés, l'un correspondant à l'événement parole, les deux autres aux deux sélections d'objets. Par Micro Fusion, ces trois Creusets vont être fusionnés en un seul Creuset, obtenant ainsi une commande complète.

Exemple 2 : Les figures 2 et 3 présentent les étapes de fusion et de défusion de l'exemple suivant : L'utilisateur sélectionne avec la souris Usair (liste dans la palette) et prononce ensuite la phrase <*Flights from Boston serving a meal*>. Ces deux événements sont fusionnés puisque ils appartiennent à une même fenêtre temporelle (Macro Fusion). L'utilisateur prononce ensuite la phrase <*Twa flights from* > tout en sélectionnant la ville <Pittsburgh>. La sélection est traitée en premier (son temps de traitement étant inférieur à la celui de la parole). La sélection <Pittsburgh> est donc fusionnée avec la requête courante (Fusion Contextuelle). Lorsque l'événement parole <*Twa Flights from*> est disponible, il est alors nécessaire de défusionner la sélection <Pittsburgh>, qui peut alors être fusionnée avec l'événement Parole, émis en parallèle de la sélection (Micro Fusion).

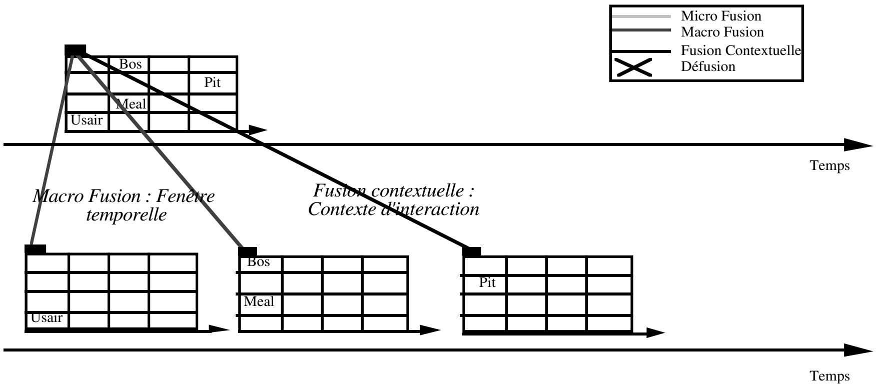


Figure 2 : 1ère étape : Macro Fusion puis Fusion Contextuelle.

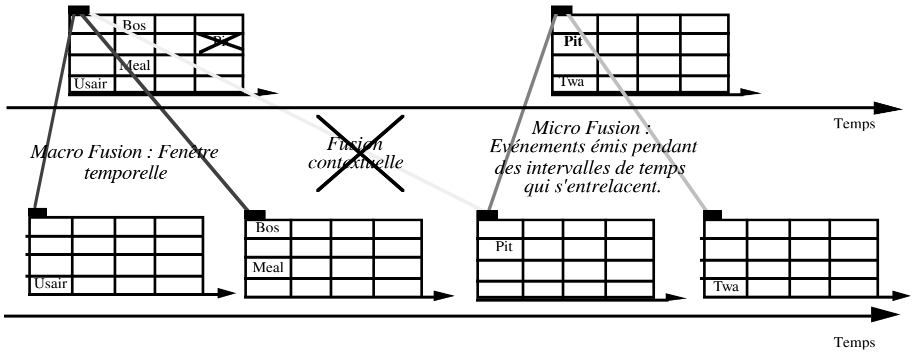


Figure 3 : 2ème étape : Défusion et Micro Fusion.

4.1.5. Conclusion

Le modèle conceptuel d'architecture appliqué est PAC-Amodeus. La fusion des informations est effectuée au niveau du composant Contrôleur du Dialogue et s'appuie sur la hiérarchie d'agents PAC qui organise ce composant.

Cette fusion est effectuée par un moteur qui manipule des informations représentées dans un formalisme unique. Ce formalisme est indépendant du domaine de l'application. De même le moteur de fusion s'appuie sur des critères tels que le temps et la structure de l'information à fusionner. Il est donc indépendant du système considéré. Son comportement peut être modifié en injectant de nouveaux critères de fusion. Afin d'obtenir un retour d'information le plus immédiat possible, sa stratégie est précoce (eager) et implique parfois de défusionner. Mais le moteur de fusion, et par conséquent le système, ne se bloquent jamais en attente d'un événement complémentaire. Cette stratégie permet en particulier la gestion d'un dialogue à plusieurs fils d'activité.

4.2. L'intégration à base de règles [Y. Bellik & D. Teil, LIMSI]

LIMSI-DRAW est une application graphique développée afin d'évaluer le système multi-modal présenté dans [Bellik 92], [Teil 91]. Elle permet de créer et de manipuler des formes géométriques élémentaires à travers un ensemble de commandes multimodales. Elle intègre trois périphériques en entrée : un système de reconnaissance vocale, un écran tactile et une souris. L'utilisation conjointe d'un écran tactile et d'une souris, qui peut sembler a priori redondante, permet d'obtenir une rapidité maximale dans la formulation des énoncés multimodaux.

4.2.1. Type de multimodalité supporté

La multimodalité supportée par le système est du type synergique, les multimodalités de type exclusive et alternée étant également couvertes. La multimodalité de type concurrent n'est pas supportée. Il est possible d'exprimer les énoncés des commandes de LIMSI-DRAW de six façons différentes : à travers la souris, à travers l'écran tactile, en combinant souris et écran tactile, en combinant parole et souris, en combinant parole et écran tactile, ou enfin en combinant parole, souris, et écran tactile.

4.2.2. Existence d'une modalité dominante

Un des objectifs visés lors de la conception du système était d'obtenir une indépendance aussi grande que possible vis-à-vis des modalités, par conséquent les différentes modalités sont traitées par le contrôleur de dialogue de manière équivalente. Les traitements effectués au cours de l'interprétation des événements sont spécifiques à chaque modalité, mais conduisent à une représentation unifiée des informations transmises au contrôleur de dialogue. Il n'existe donc pas de modalité dominante dans le système.

4.2.3. Stratégie d'intégration

La stratégie d'intégration adoptée dans le système est du type retardé. Elle a lieu au niveau du contrôleur de dialogue et consiste en deux opérations se déroulant en parallèle :

• *la fusion locale* consiste à apparier une donnée avec une référence à un argument (voir Représentation des informations multimodales),
• *la fusion globale* consiste à rassembler dans une structure commune toutes les données nécessaires à l'exécution d'une commande. C'est donc cette fusion qui produit des énoncés.

La stratégie d'exploration utilisée dans le système s'apparente à une stratégie en largeur d'abord. Il s'agit en fait d'une stratégie à interprétation multiple. En effet, les interpréteurs associés aux différentes modalités réalisent leurs tâches d'interprétation des événements de façon indépendante et autonome. Ceci a le double avantage de permettre une extension facile du système à d'autres modalités, et d'autoriser la parallélisation des processus d'interprétation. Cependant, il est évident que dans ce cas, l'interprétation d'un événement n'est pas déterministe et conduit souvent à plusieurs solutions possibles. Par exemple, la désignation d'un point appartenant à un objet peut être interprétée soit comme une désignation d'une position spatiale, soit comme une désignation de l'objet. Dans ces cas de figure, les interpréteurs fournissent au contrôleur de dialogue des informations à interprétations multiples, et c'est ce dernier qui choisit la bonne interprétation puisque disposant de toutes les informations nécessaires pour cela.

4.2.4. Critères d'intégration

La fusion locale a lieu entre deux informations satisfaisant aux conditions suivantes :
• la complémentarité logique : une des informations doit être une donnée et l'autre une référence à un argument (voir Représentation des informations),
• la compatibilité de type : le type de la donnée doit être compatible avec le type de l'argument référencé,

• la proximité temporelle : les événements ayant donné naissance à ces informations doivent avoir été produits dans des voisinages temporels proches. Cette proximité temporelle est mesurée en utilisant un seuil déterminé expérimentalement.

La fusion globale obéit à un certain nombre de règles dont voici quelques exemples :

• la structure commune doit contenir une seule information de type référence à une commande,
• une information de type donnée qui n'a pas fait l'objet d'une fusion locale est automatiquement associée au premier argument de la commande en cours satisfaisant aux trois conditions suivantes :
- il n'y a pas encore eu de référence à cet argument ayant fait l'objet d'une fusion locale,
- il n'y a pas encore eu de donnée ayant fait l'objet d'une association automatique avec cet argument,
- son type est compatible avec celui de la donnée.
Ceci suppose qu'un ordre par défaut des arguments soit défini.

• une information de type donnée qui n'a pas fait l'objet d'une fusion locale et qui n'a pu être associée à aucun argument est ignorée. Ce comportement pourrait être modifié en engageant un dialogue avec l'utilisateur.
• la fusion globale est achevée lorsque tous les arguments de la commande en cours ont été associées à des données.
• etc.

4.2.5. Représentation des informations

Si on considère que le noyau fonctionnel consiste en un ensemble de commandes activées de façon indirecte par l'utilisateur, on peut alors constater que toute activité de l'utilisateur peut être décomposée en général en une suite d'opérations de base $O_1, O_2, \ldots, O_n$ ou O_i représente l'une des opérations suivantes :

• *faire référence à une commande* : consiste à indiquer au système quelle commande doit être exécutée. Cette référence peut être complexe (s'il elle est exprimée en langage naturel par exemple) ou simple (désignation sur un menu).
• *faire référence à un argument* :consiste à indiquer au système à quel argument doit être affectée la donnée qui vient ou qui va être fournie. On peut remarquer que la parole est bien adaptée à ce type d'opération.
• *entrer une donnée* : consiste à fournir au système une valeur (objet graphique, nom de fichier, etc.) qu'il devra affecter à un certain argument de la commande en cours. Contrairement à l'opération précédente, il apparaît que le geste est très bien adapté à ce type d'opération car il permet une désignation directe à travers les périphériques de pointage.

De cette classification, découle notre formalisme de représentation. Les informations traitées par le contrôleur de dialogue sont donc de trois types : commande, argument, donnée. A chaque information est associée son type, sa modalité d'origine, la date de début de production de l'événement associé, la date de fin de production de l'événement associé, une valeur permettant d'identifier la commande, l'argument, ou la donnée, et enfin un identificateur d'état décrivant les traitements subis par l'information.

Les identificateurs d'états sont gérés par le contrôleur de dialogue. Il les affecte aux informations au fur et à mesure qu'il les traite, et selon que celles-ci aient subi une fusion locale ou pas, une association automatique ou pas, qu'elles aient été ignorées ou pas, etc. Le formalisme de représentation est donc unique (ou plutôt unifié).

4.2.6. Exemples

Exemple "mets ça ici"
Les différents processus d'interprétation et le processus de contrôle du dialogue sont censés se dérouler en parallèle bien que ce ne soit pas le cas dans LIMSI-DRAW, qui est implanté sur compatible IBM PC.

L'Interpréteur Parole (IP) reçoit l'événement relatif à la prononciation du mot "mets". Il l'interprète et produit une information de type référence à une commande (commande de déplacement d'un objet). L'Interpréteur Souris (IS) reçoit l'événement relatif au clic. Ce clic s'étant produit sur un objet, l'IS produit une information à double interprétation : une donnée de type position spatiale et une donnée de type objet graphique. L'IP reçoit l'événement relatif à la prononciation du mot "ça". Il l'interprète et produit une information de type référence à un argument (argument OBJET). L'IS reçoit l'événement relatif au second clic. Si ce clic s'est produit sur un objet, l'IS produit une information à double interprétation, sinon une information à interprétation unique (donnée de type position spatiale), est générée. L'IP reçoit l'événement relatif à la prononciation du mot "ici". Il l'interprète et produit une information de type référence à un argument (argument POSITION).

Le Contrôleur de Dialogue (CD) traite les informations provenant des interpréteurs selon l'ordre chronologique réel des événements correspondants. Pour traiter une information, il utilise un ensemble de règles dégagées après analyse de toutes les configurations possibles qui peuvent se présenter dans la file d'attente des informations, et dépendant principalement du type de l'information courante et de l'information précédente, de l'état de cette dernière, de la compatibilité de types (entre arguments et données), et de la proximité temporelle des deux informations. Lorsque tous les arguments de la commande ont été valués, il transmet alors la structure commune contenant les valeurs des arguments à l'adaptateur du domaine qui déclenche l'exécution de la commande.

Autres exemples d'interactions
L'utilisateur prononce le mot "rectangle" tout en cliquant avec la souris et en pointant avec son doigt sur l'écran tactile. Les deux points ainsi désignés représentent les extrémités de la diagonale.

L'utilisateur prononce le mot "cercle". Il prononce ensuite le mot "bord" tout en désignant un point sur l'écran tactile. Il prononce le mot "centre" tout en appuyant sur le bouton de la souris. Un cercle défini par les deux points précédents s'affiche. L'utilisateur peut alors faire varier le centre du cercle en bougeant la souris. Il terminera en relâchant le bouton.

L'utilisateur prononce la phrase "triangle vert" en désignant les trois sommets par la souris et/ou l'écran tactile. En maintenant son doigt sur l'écran tactile lors de la désignation du troisième sommet, il pourra faire varier la forme du triangle en jouant sur ce dernier sommet et en faisant glisser son doigt sur l'écran.

4.2.7. Conclusion

LIMSI-DRAW a été une expérience enrichissante. Cependant, certains problèmes demeurent encore non résolus, tels que la gestion du dialogue à plusieurs fils (basculement d'un énoncé à un autre) et du dialogue concurrent (simultanéité dans la formulation d'énoncés indépendants), ainsi qu'un traitement plus modulaire des sorties. Ceci fait partie de nos préoccupations actuelles car nous pensons que les interfaces multimodales doivent se situer au dessus des interfaces classiques qui souvent arrivent à résoudre ces types de problèmes.

4.3. La fusion des événements dans l'interface du modeleur déclaratif [N. Vigouroux, E. Brison, V. Gaildrat, O. Ballet IRIT]

Nous allons présenter le modèle d'interprétation des énoncés multimodaux mis en œuvre dans l'interface d'un modeleur déclaratif de scènes travaillant dans l'espace tridimensionnel [Gaildrat, 1993]. Les limitations d'utilisation tels que le placement d'objets dans la scène, l'acquisition de données non géométriques (textures, matériaux, ...) des modeleurs graphiques amènent à l'utilisation de la modélisation déclarative. Cette modélisation déclarative offre à l'utilisateur la possibilité de décrire une scène à modéliser par le biais de commandes purement orales et écrites ou combinées avec de la manipualtion directe. Par ce biais, l'utilisateur dispose d'ordres lui permettant d'effectuer les modifications à opérer sur les différents objets d'une scène.

Cette application où l'activité est de créer, d'assembler, de modifier les propriétés des éléments de la scène et d'effectuer sur eux des changements spatiaux offre un cadre expérimental intéressant pour l'étude de la multimodalité. La résolution des références intermodales dans le cadre de cette application fait appel à :
• des références spatio-temporelles ("Mets la bouteille sur la table"),
• des références au discours (résolution des anaphores, des ellipses…),
• des références pragmatiques résultant de la manipulation directe,
• des coréférences du langage naturel— forme écrite ou orale— et de la désignation directe.

Les commandes du modeleur les plus usitées permettent d'interagir au moyen de la souris tout en complétant à la voix la commande (cas d'une complémentation d'une commande par une autre modalité) ou de modifier une commande en cours d'exécution (par exemple, sur les propriétés des objets, luminosité, couleur, représentation spatiale, …).

Le modeleur ajoute une dimension supplémentaire à considérer, celle de l'animation d'objets qui induit des représentations continues.

En nous limitant au problème des entrées dans le cas de cette présentation, nous distinguons les trois fonctions suivantes :
• l'acquisition des informations fournies par l'usager (a)
• leur interprétation pour obtenir une commande complète (b)
• et la gestion de cette commande à travers le système de dialogue (c).

Seul le point b sera abordé ici. Les commandes utilisées réfèrent à trois unités de base : à une action, aux arguments nécessaires à l'exécution de cette action et aux relations entre objets (par exemple, -posé-sur-, -collé-contre-, ...).

Tout d'abord nous décrivons la forme de la représentation d'une commande d'actions. Le mode de représentation est indépendant des moyens mis en œuvre pour les interactions —souris, parole, clavier. La structure retenue est une structure de type informationnelle que l'on peut schématiser de la manière suivante :

< actions, objets, lieu, lien >

Par exemple, 1. efface le; 2. mets ça là; 3. aligne les verres sur la table, etc. Les niveaux de représentation subordonnés permettent de spécifier complètement l'énoncé de la commande en instanciant les tr*aits*. Ils sont définis de manière à pouvoir intégrer plusieurs modes d'interaction de manière transparente pour l'utilisateur. Dans les exemples (1) et (2), la commande n'est pas complète. L'objet de référence de /le/ ou du /çà/ de même que la position de référence de /là/ doivent être satisfaites par le contexte (référence intermodalité). Les représentations des objets et des positions permettront de compléter ces formes verbales par des informations tirées du contexte (i.e. l'historique du dialogue, les informations véhiculées par le mode gestuel). Les objets et les positions désignées sont attachés à des champs qui sont remplis lors de l'intégration des références intermodalités.

4.3.1. Type de multimodalité supporté

Les types de multimodalité supportés par le système sont de type synergique, exclusif et alterné.

4.3.2. Existence d'une modalité dominante

Un des objectifs visés lors de la conception du système d'intégration des modalités était d'obtenir une indépendance aussi grande que possible vis-à-vis des modalités. Il n'existe donc pas de modalité dominante dans le système de fusion. Une de nos contraintes fortes que nous nous étions imposées lors de la conception était d'élaborer un mécanisme de fusion aussi indépendant que possible de la modalité. Ceci nous a conduit à l'élaboration d'une structure de représentation unifiée des informations transmises au noyau fonctionnel.

4.3.3. Critères de fusion

Les critères de fusion utilisés dans le noyau d'intégration relèvent essentiellemnet de :
- contraintes temporelles,
- de connaissances pragmatiques,
- de connaissances a priori sur l'utilisation des média (a) ;
- et des connaissances sur l'état du dialogue (b) .

(Les critères (a) et (b) ne sont pas utilisés dans le mécanisme d'intégration des références intermodales),

4.3.4. Exemples

Exemple "mets ça là "

L'expression de cette commande est différente suivant les usagers. Il peut parfois apparaître le choix d'une modalité dominante :

• la désignation 2D pour les habitués des interfaces graphiques; les formes de surface I et II sont généralement des énoncés monodaux; la forme III est caractéristique d'un usager non habitué qui traduit sa logique de formulation.

• le langage oral où l'expression (III ou IV) sera plus familière pour des adeptes du langage naturel. Ces formes peuvent être énoncées de manière multimodale.

Cette prépondérance d'une modalité est un paramètre que nous considérons dans l'interprétation pour la résolution des coréférences.

Exemple 2 - Les commandes qui génèrent un entrelancement des E/S

Les commandes "dynamiques" ou "progressives" nécessitent un axe supplémentaire d'interprétation face à l'enchaînement (aspect continu) des interactions réalisées. Cet axe concerne l'animation des objets de la scène qui induit des représentations spatio-temporelles de ces objets dans la scène. Une commande "dynamique" se compose d'un ordre (par exemple, "déplacer") et des paramètres nécessaires à cet ordre (objet, direction). Mais à la différence des commandes "instantanées", l'application de la commande est réalisée de manière continue. A chaque instant, l'utilisateur peut modifier la vitesse d'exécution, la direction de la commande ..., ou stopper l'exécution, tout ceci au moyen de nouvelles interactions textuelles ou gestuelles qui viennent modifier le contexte de réalisation de l'interaction précédente.

Soit l'exemple suivant :

• A partir de la commande "déplacer l'objet X vers la droite", l'objet est déplacé avec une vitesse relativement réduite.

• L'utilisateur choisit de cliquer sur le bouton "+" de l'interface graphique : la vitesse de déplacement de l'objet est augmentée.

• L'objet dépasse la position souhaitée, l'utilisateur précise "peu" et clique sur la flèche droite : la direction et la vitesse sont ainsi modifiées.

• Quand l'objet atteint la bonne position, l'utilisateur l'immobilise en énonçant "stop", ce qui a pour effet de terminer l'exécution de cette commande.

Ces interactions doivent ainsi être considérées, soient comme participant à l'élaboration d'un nouvel énoncé multimodal, soient comme étant des modifications d'une commande en cours d'exécution. L'interprétation doit donc savoir qu'une commande "dynamique" est en cours d'exécution afin de correctement prendre en compte les interactions reçues.

4.3.6. Conclusion

L'interprétation de l'énoncé est réalisée par un mécanisme qui manipule des connaissances représentées dans un formalisme unique. Ce formalisme est indépendant du domaine de l'application. De même ce mécanisme de fusion qui s'appuie sur des critères tels que le temps, la structure de l'énoncé, semble a priori indépendant des modalités d'entrée.

La plate-forme (interface et modeleur) est en cours de réalisation. Pour le cas des commandes continues il est nécessaire de gérer les problèmes d'exploitation simultanée des modalités à la fois en entrée et en sortie (i.e. l'étude des relations entre les entrées et les sorties). Ce travail nécessite tout d'abord d'identifier les interactions à caractère continu dans le modeleur et de s'interroger sur le "droit" qu'a l'utilisateur d'interrompre la tâche en cours. Ceci soulève le problème de la prise en compte d'actions rapides de la part de l'utilisateur au sein même de l'interface.

4.4. L'intégration à un niveau conceptuel [F. Azémard & J.P. Macchion, IRIT]

Le modèle exposé ci-après ne fait pas à l'heure actuelle l'objet d'une implémentation complète. Cependant, il est plus particulièrement élaboré pour des applications de formation ayant pour cadre un environnement graphique. Nous considérons une modalité langagière sous sa forme orale et écrite et des clic-souris.

4.4.1. Type de système mutimodal

Notre système est hybride, nous envisageons a priori tous les types de multimodalité. La multimodalité concurrente n'est pas considérée pour la construction de plusieurs énoncés. Mais elle intervient localement à un énoncé pour privilégier une association plutôt qu'une autre. Par exemple, pour un clic-souris et un pronom dans une phrase, considérons que le premier est effectué en même temps que le second est prononcé. Cette proximité temporelle peut jouer en faveur d'une association. Or, une connaissance sur l'utilisateur peut indiquer que ce pronom est employé indépendamment d'un clic-souris. Nous considérons alors qu'il y a concurrence locale et l'association n'est pas faite.

4.4.2. Existence d'une modalité dominante

Le langage naturel ou pseudo-naturel est la modalité dominante. La modalité parole sert de pivot à l'interprétation. La modalité gestuelle, par le biais de clics-souris, vient compléter des éléments manquants nécessaires à la construction d'un énoncé.

4.4.3. Stratégie d'intégration

Les événements sont amenés jusqu'à une représentation commune. Le niveau d'abstraction est d'ordre sémantique. L'intégration est donc réalisée au sommet de la pyramide ARCH par le seul contrôleur de dialogue. La stratégie "lazy" d'intégration est ensuite, le fait de règles à déclenchement réflexe.

4.4.4. Représentation des informations

Le formalisme de représentation adopté est unique, il est très proche des graphes conceptuels [Sowa 84]. Un graphe traduit dynamiquement ce qui a été compris d'un événement. Nous parlons de graphe de compréhension. Les mots d'un événement parole ont au travers d'un lexique une ou plusieurs représentations conceptuelles. A un mot qui n'est pas concept de l'application, correspond un concept générique complété d'un marquage lexical. A un clic-souris est associé un graphe à un seul sommet qui identifie l'objet cliqué.

4.4.5. Critères d'intégration

Les critères d'intégration utilisés sont :

• la proximité temporelle ; nous n'utilisons pour l'instant qu'une file d'attente des événements, mais ces derniers sont datés.

• la complétude d'une commande ; les règles à déclenchement réflexe sont sensibles à la présence de sommets non-instanciés dans un graphe. L'intégration consiste à associer une valeur à chaque sommet non-instancié du graphe de compréhension. L'ordre de passage est donné par la position des mots dans l'événement parole.

• le typage au niveau sémantique ; les sommets à fusionner doivent être du même type ou pouvoir être réduits au même type par spécialisation.

• les connaissances liées au contexte du dialogue ; en particulier, des connaissances sur la manière dont l'utilisateur emploie les types de modalités interviennent dans les prémisses de certaines règles.

4.4.6. Exemples

Nous présentons au travers des exemples suivants les étapes successives qui conduisent des premiers graphes jusqu'au graphe final de compréhension d'un énoncé. Ces étapes sont au nombre de trois, la génération des graphes conceptuels relatifs à chaque modalité, puis la construction du graphe de compréhension de l'événement parole, et enfin l'intégration au moyen des règles réflexes. Détailler les mécanismes de *jointure* et *filtrage* qui nous permettent d'obtenir le graphe d'un énoncé serait trop long ici, nous renvoyons le lecteur à [Azémard 92a] et [Azémard 92b].

Les graphes doivent se lire de la manière suivante : [*xclasse] : est une variable qui capture uniquement une instance de la descendance de la classe spécifiée, et x est un chiffre permettant de distinguer plusieurs variables d'une même classe. {marquage} : marquage lexical de premier niveau, pronom, article défini, déictique ou adverbe ...

Exemple "mets ça là" accompagné de deux clics souris

(a) Phase de génération des premiers graphes conceptuels pour les modalités

• Souris ; A un clic souris correspond un objet de l'application, ici les graphes sont des sommets isolés. Soient [Objet14] pour le clic1 et [Lieu21] pour le clic2.

• Modalité parole ; le lexique fournit les graphes suivant :

mets :	[Mettre0] - (objet) -> [*1objet] (lieu) -> [*1lieu]
ça :	[*2objet{déictique}]
là :	[*2lieu{adverbe}] ou [*3objet{pronom}] ou [*4objet{article défini}]

(b) Phase de construction du graphe de compréhension de l'événement parole

[*4objet{article défini}] est écarté par une règle syntaxique spécifiant qu'un article doit précéder un nom. Dans la hiérarchie des classes, OBJET est ancêtre de LIEU. Plusieurs associations sont donc possibles. Nous observons quatre cas :

1 : [*1objet] ≡ [*2objet{déictique}] et [*1lieu] ≡ [*2lieu {adverbe}]

2 : [*1objet] ≡ [*2objet{déictique}] et [*1lieu] ≡ [*3objet{pronom}]

3 : [*1objet] ≡ [*2lieu{adverbe}] et [*1lieu] ≡ [*2objet{déictique}]

4 : [*1objet] ≡ [*3objet{pronom}] et [*1lieu] ≡ [*2objet{déictique}]

Pour choisir, nous employons une mesure d'éloignement sémantique. Cet éloignement est la somme des spécialisations à réaliser sur l'ensemble des sommets concernés par la jointure dirigée pour rendre notre graphe conceptuel connexe. Ici, le cas 1 est retenu, l'éloignement est égal à zéro. Pour tous les autres, il est supérieur ou égal à 1. En effet, il y a au moins une spécialisation à faire pour passer de objet à lieu. Le graphe de compréhension de l'événement parole, après jointure dirigée, est donc le suivant :

[Mettre0] - (objet) -> [*1objet ≡ *2objet{déictique}] (lieu) -> [*1lieu ≡ *2lieu {adverbe}]

(c) Intégration au travers de règles

L'intégration est réalisée au travers de règles du type :

SI	*élément de File_Evénement = *évt_clic_souris*	
	*élément de File_Sommet = *Sommet*	déclencheur
	*type de *Sommet = non_instancié*	
	*marquage de *Sommet = "déictique"*	
	(mode_synergique)	
ALORS	*(tentative_intégration *évt_clic_souris *Sommet)*	
	*type de *Sommet <- instancié*	

Après l'application de la règle précédente le graphe de compréhension de l'énoncé multimodal renvoyé par le contrôleur de dialogue est :

```
[Mettre0] -
  (objet) -> [Objet14]
  (lieu)   -> [Lieu21]
```

L'objet désigné par le premier clic souris sera placé à l'endroit pointé par le deuxième clic.

Autre exemple où l'intégration n'est pas toujours Fusion:

Il faut parfois observer le contexte pour lever des ambiguïtés. Par exemple, dans un même événement parole, un terme ambigu peut être associé à un ou plusieurs clics souris et un autre terme verra son ambiguïté levée par une recherche dans l'historique du dialogue.

Considérons l'exemple suivant, où l'utilisateur vient juste d'ouvrir une fenêtre. Il prononce alors "mets ce couple dans la fenêtre" et l'accompagne de deux clics. Sur des bases morphologiques, le premier clic serait associé à couple et le second à fenêtre. Or il faut comprendre "mets-y ce couple". Au niveau conceptuel il apparaît qu'un couple est formé de deux éléments. Ce sont ces deux éléments que l'utilisateur désigne successivement à l'écran.

Les graphes conceptuels des clics souris identifient deux objets [Objet17] et [Objet23]. Le graphe de compréhension de l'événement parole est le suivant :

```
[Mettre4] -
     (objet) -> [*1couple{déictique}] -
                      (élement1) -> [*1objet{déictique}]
                      (élément2) -> [*2objet{déictique}]
   (lieu)  -> [*1fenêtre{article_défini}]
```

Le marquage lexical "déictique" s'est étendu aux composants du couple. L'emploi des déictiques est généralement associé à la multimodalité synergique. De même, la référence à un objet à l'aide d'un article défini indique très souvent qu'il s'agit d'un objet précédemment cité.

Le graphe conceptuel complètement instancié sera le suivant :

```
[Mettre4] -
  (objet) -> [couple12] -
                 (élement1) -> [Objet17]
                 (élément2) -> [Objet23]
  (lieu)  -> [Fenêtre3]
```

Les deux objets désignés à l'écran par l'utilisateur sont donc déplacés dans la fenêtre qu'il venait d'ouvrir.

4.4.7. Conclusion

Nous avons opté pour un unique formalisme de représentation d'ordre conceptuel. En effet, l'intégration événementielle ne se limite pas à la seule mise en correspondance des événements pour une fusion. Il s'agit aussi de prendre en compte toutes les connaissances que manipule le contrôleur de dialogue. C'est pourquoi, en nous appuyant sur une modalité langagière dominante, nous placons sur un pied d'égalité, intégration, recherche dans l'historique du dialogue, recherche dans la base de faits et reconnaissance de plan. De plus, lorsque les événements sont fusionnés dès le premier niveau, ils n'existent plus en tant qu'événements dans les niveaux supérieurs. Ils sont phagocytés, et il faut déstructurer si les associations ne sont pas bonnes. Nous pensons que l'événement doit exister du plus bas niveau jusqu'au contrôleur de dialogue.

4.5. Modélisation connexionniste pour l'interaction homme-machine multimodale [J.-C. MARTIN & D. BÉROULE, LIMSI]

Les éléments de modélisation décrits ci-dessous font partie d'une architecture homogène pour la communication homme-machine en cours de développement au LIMSI et faisant intervenir un principe de détection de coïncidences [Béroule 92]. On s'intéresse ici à l'utilisation de ce modèle pour l'intégration d'informations multimodales [Martin, Béroule 92]. Ces éléments de modélisation et de réflexion ne sont pas encore intégrés dans une application significative.

4.5.1. Modèle d'architecture conceptuelle

L'architecture globale fait intervenir des modules correspondant à différentes modalités ou à différentes analyses ou synthèses. Tous les modules font intervenir les mêmes représentations et mécanismes (propagation de signaux, apprentissage à partir d'exemples, facilitation de la propagation...). Leur rôle et leurs interactions dépendent uniquement de leur connectivité et des paramètres qui leur sont attachés (durée des signaux, seuil, ...) et non de sources de connaissances spécialisées. L'architecture globale fait intervenir des modules à différents niveaux d'abstraction. L'intégration peut avoir lieu à chacun de ces niveaux.

4.5.2. Représentation des informations

La représentation des connaissances utilisée est de type connexionniste : elle fait intervenir des unités de traitements élémentaires (ayant quelques paramètres : seuil, activation) connectées entre elles par des liens. Des unités appelées *détecteurs de faits* représentent des messages élémentaires perçus par la machine ou des résultats d'analyses plus complexes.

L'apparition de plusieurs faits émis en même temps ou proches temporellement est détectée par une unité interne (appelée aussi unité contextuelle ou DC pour dépendante du contexte) vers laquelle converge des signaux émis par les détecteurs de faits correspondants. L'activation de cette unité interne correspond alors à la perception d'un *événement*.. Des *séquences d'événements* peuvent être reconnues grâce à la propagation de signaux sur des liens reliant des unités internes entre elles. Les événements possibles sont rapidement pris en compte grâce à leur préactivation par leur contexte d'apparition (unité interne associée à l'événement précédant dans la séquence à reconnaître).

4.5.3. Type de système multimodal

On peut envisager d'utiliser temporellement les modalités en séquentiel ou en parallèle. Les différents types de multimodalité sont possibles selon la connectivité du réseau et les mécanismes utilisés :

• Une unité interne peut être reliée à des détecteurs associés à une ou à plusieurs modalités.
• Les unités internes d'un chemin sont toutes reliées à la même modalité ou non.
• La propagation peut se faire simultanément sur plusieurs chemins (d'un ou plusieurs modules) ou non (inhibition entre les chemins).

Exclusif **:** les unités d'un même chemin sont toutes reliées à des détecteurs d'une même modalité. La propagation des signaux ne se fait que sur un chemin à la fois (le chemin associé à l'énoncé en cours de reconnaissance).

Alterné **:** cette fois, le chemin associé à un énoncé peut faire intervenir plusieurs modalités. Mais chaque unité du chemin n'est reliée qu'à une modalité à la fois. La propagation des signaux ne se fait que sur un chemin à la fois.

Concurrente : les unités internes d'un même chemin peuvent être reliées à plusieurs modalités. La propagation des signaux peut se faire sur plusieurs chemins à la fois (les chemins associés aux énoncés en cours de reconnaissance).

4.5.4. Existence d'une modalité dominante

Les éléments de modélisation sont suffisamment génériques pour qu'il y ait une modalité dominante ou non selon l'application et les modalités disponibles.

Une modalité dominante peut intervenir de différentes manières :

• en orientant de manière déterminante les chemins choisis (début de chemin relié à des mots) dans un module multimodal,

• en influençant, à partir d'un module monomodal associé à cette modalité dominante, d'autres modules associés à d'autres modalités (facilitation de la reconnaissance).

4.5.5. Stratégie d'intégration

L'exploration des différentes hypothèses est réalisée en parallèle. En cas d'ambiguïté on peut choisir soit la première hypothèse reconnue, soit l'hypothèse la plus probable (niveau d'activation le plus élevé). Selon les valeurs affectées aux différents paramètres (délais des liens, durée des signaux), l' exploration peut avoir lieu en largeur (les signaux envoyés par les hypothèses arrivent en même temps au module suivant qui peut alors choisir l'hypothèse la plus activée), ou en profondeur (choix d'un chemin, propagation le long de ce chemin, en cas de non activation complète du chemin, rebouclage pour activer un autre chemin).

4.5.6. Critères d'intégration

Les critères à utiliser dépendent fortement de l'application et des modalités utilisées. Le critère de proximité temporelle est réalisé grâce à la détection de coïncidences dans les unités internes.

L'intégration effectuée dépend aussi des connexions existantes reliant les unités et les modules. Une unitévers laquelle convergent des liens reliés à des unités représentant des positions pourra détecter des coïncidences spatiales. Les chemins existants peuvent aussi représenter des contraintes syntaxiques.

L'intégration dépend des signaux reçus par une unité et donc des activités des autres unités qui ont envoyé ces signaux. Ces unités peuvent représenter un contexte, ou un chemin complètement activé.

L'intégration s'effectue en même temps dans plusieurs éléments ayant la même structure de données (les unités internes) vers lesquels convergent des liens reliés à d'autres unités. Les mécanismes utilisés sont : la propagation des signaux, la sommation de ces signaux dans les unités internes.

4.5.7. Exemples

Exemple du "mets ça ici"

Une commande utilisant la reconnaissance de parole et la souris sera identifiée grâce à une propagation d'activité dans le réseau de la figure 2 de notre article dans les actes d'IHM'92 (page 45).

Cet exemple fait intervenir trois modules : "Reconnaissance de parole", "Souris", et "Commandes multimodales".

Le module "Reconnaissance de parole" contient trois chemins aboutissant aux détecteurs de mots "mets", "ça" et "ici". Le module "Souris" contient un détecteur "Clic".

Le module "Commandes multimodales" contient un chemin composé de trois unités internes DC1-DC2-DC3 aboutissant au détecteur de fin de chemin "mets ça ici".

L'unité interne DC1 est connectée au détecteur "mets".

L'unité DC2 est connectée à DC1 et aux détecteurs "ça" et "clic".

L'unité DC3 est connectée à DC2 et aux détecteurs "ici" et "clic".

Lors de la reconnaissance, l'activation du détecteur du mot "mets" se propage vers la cellule DC1, qui envoie un signal contextuel à DC2. Si ce signal coïncide grossièrement avec l'activation de la souris et la reconnaissance du mot "ça", DC2 propage à son tour son activation vers DC3. Le guidage de la propagation le long du chemin de cellules contextuelles se poursuit ainsi jusqu'au détecteur de la commande "mets ça ici". La durée des signaux intégrés par chaque DC permet de pallier les variations temporelles d'occurrence des impulsions provenant des différentes modalités.

Nous avons uniquement décrit un moyen de reconnaître syntaxiquement la commande multimodale mets ça ici. Pour la traiter, il faudra représenter les objets graphiques de l'éditeur, faire le lien entre la localisation qui coïncide avec le mot ça et l'objet désigné...

Autre exemple

La multimodalité concurrente est possible en propageant en parallèle des signaux sur plusieurs chemins. L'utilisateur pourrait émettre en même temps : un message vocal "*éjecter disquette*" alors qu'il est en train de taper une autre commande au clavier "*aide éditeur*". Les messages associés à ces deux énoncés indépendants provoquent la propagation de signaux sur des chemins (et même des modules) séparés. La coïncidence temporelle n'étant pas le seul critère utilisé pour fusionner des messages, des chemins pré-existants peuvent orienter la fusion et permettre la reconnaissance d'énoncés simultanés et indépendants.

4.5.8. Conclusion

Le traitement en temps réel et l'intégration de données multimodales, la prise en compte de distorsions temporelles, la possibilité de faire coopérer plusieurs modalités font partie des besoins d'un système de dialogue homme-machine multimodal. Nous avons proposé une approche connexionniste qui nous semble pouvoir répondre à ces besoins. Nous étudions actuellement les possibilités de prise en compte d'éléments variables dans des réseaux comme l'association dynamique d'une signification à un message perçu dans une modalité en fonction d'un message perçu dans une autre modalité.

4.6. ICPplan : dialogue multimodal pour la conception de plans architecturaux [Bourguet M.-L., ICP]

ICPplan [Bourguet 92b] est une application de dessin multimodale (parole et gestes de désignation) qui offre les outils nécessaires à la réalisation de plans architecturaux. Le choix d'une application où l'activité est de créer des objets en nombre variable, d'assembler, de modifier et de positionner ces objets dans un espace, offre un bon terrain d'étude pour les références multimodales : références au discours, anaphores et ellipses (Ex. : Dessine une fenêtre. Agrandis la) ; références graphiques et spatiales (Ex. : Mets la chaise à côté de la petite table) ; références gestuelles (manipulation directe) ; et coréférences parole et geste (Ex. : Dessine une porte sur ce mur + désignation gestuelle).

4.6.1. Architecture générale

L'architecture générale du système utilise une décomposition en couches qui permet de répartir la complexité du logiciel dans plusieurs niveaux d'abstraction. Elle s'articule autour des trois composants suivants : un gestionnaire des événements, un analyseur multimodal et un contrôleur de dialogue.

• *Le gestionnaire des événements* est chargé de gérer les événements que lui envoient le serveur de parole d'une part (le speech manager) et le serveur graphique d'autre part (le window manager). Ces événements sont issus des différents médias disponibles : le microphone, la souris et le clavier. Ils sont caratérisés par une date et un média, et véhiculent des informations modales. Ils ont été définis de la façon suivante :

dK : pour signaler qu'un bouton de la souris a été enfoncé ;
fK : lorsque le bouton de la souris est relâché ;
dP : pour signaler que l'utilisateur a commencé à parler ;
fP : signale la fin du message oral (l'utilisateur a cessé de parler) ;
REC : le résultat du système de reconnaissance, donc la phrase prononcée, est disponible.

• *L'analyseur multimodal* est chargé d'analyser les informations véhiculées par les événements en utilisant des connaissances multimodales (connaissances propres à chacun des modes de communication). En sortie, il est chargé d'envoyer au système de synthèse de la parole une chaîne de mots à prononcer et de communiquer à l'interface graphique des commandes de présentation de l'écran.

• *Le contrôleur de dialogue* est un module central dans l'architecture de l'interface. Il a à sa disposition plusieurs sources de connaissance : modèle du dialogue (pour contrôler l'interaction), modèle de la tâche et des objets (pour interpréter les énoncés), et il construit au fur et à mesure de l'interaction trois historiques : un historique des tâches et deux historiques des objets.

4.6.2. Représentation des informations

Le contrôleur de dialogue accède à trois types de représentation :

• *les structures événementielles*, générées par le gestionnaire des événements. Elles contiennent les données suivantes : une date, un média et un "contexte multimodal". Ce contexte multimodal doit rendre compte en permanence de l'activité de l'opérateur. Il peut prendre les valeurs : que l'utilisateur est en train de parler (dP, fP) ou souris en action (dK, fK), définissant ainsi des fenêtres temporelles à l'intérieur desquelles les événements seront fusionnés.

• *les structures informationnelles*, générées par l'analyseur multimodal. Elles constituent une représentation commune de l'information, indépendante du mode de communication (il n'y a donc pas de "modalité dominante" dans ICPplan). Cette structure est du type casuelle, contenant en particulier une action et les paramètres de celle-ci.

• *les structures de l'intervention*, générées également par l'analyseur multimodal. Elles rendent compte de la valeur de l'intervention dans le cadre du dialogue (commande, demande d'aide, réitération ou correction de l'action précédente, etc.).

4.6.3. Stratégies et Critères d'intégrations

L'intégration des modalités (ou fusion des informations) est opérée dans le modèle de la tâche (MT), donc à un haut niveau d'analyse (niveau pragmatique). Nous proposons un modèle de la tâche qui est un modèle actif, c'est-à-dire capable de raisonner sur les données. Cette façon de concevoir le modèle de la tâche permet d'assurer une grande indépendance entre le contrôleur de dialogue (CD) et l'application.

Mais le processus d'intégration peut être aussi déclenché à très bas niveau, sur des critères de proximité temporelle :

• *si deux événements consécutifs* sont dissociés dans le temps, c'est-à-dire s'ils n'appartiennent pas à la même fenêtre temporelle (leurs contextes multimodaux respectifs sont vides), la fusion des informations est dictée par les besoins de la tâche (niveau pragmatique). Elle peut se faire grâce aux historiques du dialogue (Ex. : l'utilisateur sélectionne un objet puis dit "détruis"), ou par l'attente des informations à venir (avec éventuellement un message de la machine) (Ex. : l'utilisateur dit "détruis ça" puis sélectionne un objet). Dans les deux cas, c'est le modèle de la tâche qui guide la fusion en fonction de ses besoins d'information ;

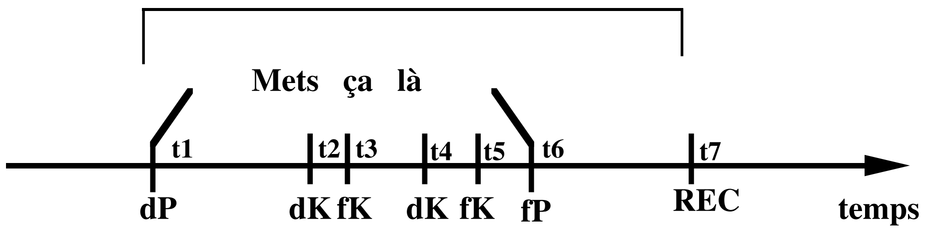
• *si deux événements sont simultanés* (si en fait ils appartiennent à la même fenêtre temporelle), la fusion est dictée par le contrôleur de dialogue, sur des critères de proximité temporelle (Ex. : l'utilisateur dit "pivote les triangles" tout en sélectionnant par le geste un ensemble d'objets).

Dans tous les cas, lorsque deux informations doivent être fusionnées pour constituer un message, le système est capable d'effectuer l'intégration (système multimodal composé ou synergique). Par contre, lorsque deux informations simultanées surviennent, le CD n'est pas capable de les dissocier, il impose toujours au MT de tenter la fusion. Ce dernier peut considérer deux commandes indépendantes, à condition que l'une d'entre elles au moins détermine complètement une tâche. Si ce n'est pas le cas, la fusion est opérée coûte que coûte, et le système est amené alors à gérer des cas de redondance, de complémentarité et de contradiction entre les informations. Autrement dit, la stratégie de gestion des événements et de fusion des informations empêche ICPplan d'être utilisé dans le style multimodal concurrent. Cette limitation est imposée aussi par le modèle du dialogue qui ne peut gérer qu'une seule tâche à la fois [Bourguet 92a, c].

4.6.4. Exemples

Exemple "mets ça là"

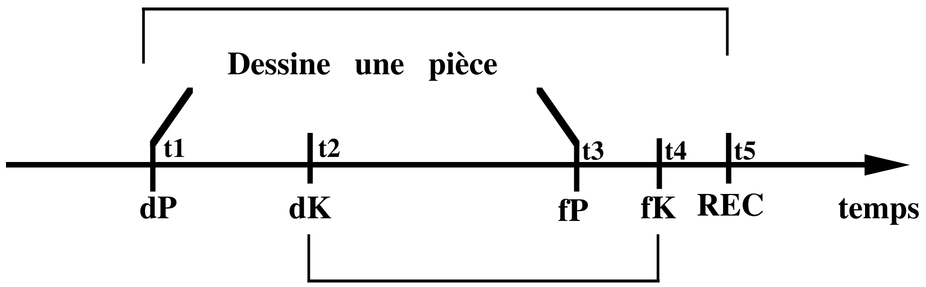
L'utilisateur prononce la commande "Mets ça là" tout en désignant d'abord un objet, puis un emplacement de l'écran. La séquence d'événements correspondante est représentée dans la figure suivante :



Dans cet exemple, les clics souris surviennent tous deux dans un contexte multimodal l'utilisateur est en train de parler, l'énoncé oral n'étant pas encore connu du contrôleur de dialogue. Celui-ci décide de conserver les informations gestuelles dans un schéma de tâche général, et de se mettre en attente des informations verbales pour déterminer la tâche à exécuter. Lorsque l'événement REC survient, la structure informationnelle qui lui est associée est intégrée dans le schéma de tâche actif où est opérée la fusion des informations gestuelles et verbales.

Autre Exemple

L'utilisateur prononce la commande dessine une pièce tout en parcourant un espace sur l'écran pour préciser à la fois la taille et la position de la pièce.



Comme dans l'exemple précédent, l'information gestuelle qui est obtenue dans un contexte multimodal non vide est conservée dans un schéma de tâche générique. La tâche à exécuter reste indéterminée. A l'arrivée de l'événement REC le CD impose au MT d'intégrer l'ensemble des informations sur un critère de proximité temporelle mis en évidence par le contexte multimodal. De cette façon, le MT prend en compte la zone de l'écran désignée qui fixe à la fois une taille et une position pour l'objet pièce. Le message multimodal fractionné sur deux modes de communication est reconstitué.

4.6.5. Conclusion

ICPplan constitue une "plate-forme multimodale", c'est-à-dire qu'il offre les outils nécessaires à la mise en œuvre d'un dialogue multimodal. Il existe en fait à l'heure actuelle peu de systèmes de communication qui intègrent des concepts ergonomiques pouvant les rendre plus efficaces et plus fiables. Mais des conceptions d'interface doivent être faites malgré tout, sans attendre les résultats théoriques, pour amorcer dès maintenant les évaluations ergonomiques. C'est en particulier ce que nous montre ICPplan, à travers le problème de la coréférence entre les modes et le problème du parallélisme des médias (effets d'anticipation, de redondance, de conflit, etc.).

5. Intégration des modalités : synthèse [T. Baudel, LRI & N. Carbonell, CRIN-INRIA]

Nous nous bornerons, d'une part, à situer les approches présentées par les participants (cf. le paragraphe 4) dans l'espace de caractéristiques défini au cours des discussions (cf. le paragraphe 3), d'autre part, à formuler quelques observations personnelles sur les problèmes que soulève, lors de l'interprétation des interventions multimodales de l'usager, l'intégration/fusion de modalités diverses comprenant, notamment, la parole naturelle (i.e. l'utilisation orale de la langue naturelle) et le geste (via la souris ou un écran tactile).

5.1. Tableau synoptique des approches proposées au cours de l'atelier

Caractéristique Système	Type de multi-modalité	Modalité dominante	Intégration	Exploration	Critères d'intégration	Structure de données
	(1)	(2)			(3)	
Nigay	S C	n	Précoce	Profondeur	t l c	Intermédiaire
Bellik	S	n	Paresseuse	Largeur	t l c *i*	Dialogue
Vigouroux	S	d (p ou m)	Paresseuse	-	t l c [h] *i*	-
Azémard	S [C]	y (p)	Paresseuse	Largeur	t l h	Dialogue
Martin	S C	n	-	Au choix	r	Répartie
Bourguet	S	y (p)	Précoce	Largeur	t l c h	Répartie

Les valeurs non spécifiées ou non significatives sont signalées par un -

Notes :

(1) S : synergique (alternée ou non), C : concurrente. Bien naturellement, tous les systèmes proposés offrent des possibilités de multi-modalité synergique.
(2) y : interaction basée sur (p) un modèle de dialogue oral, (m) un paradigme de manipuation directe.
d : dépendant du mode d'interaction/de dialogue courant,
n: modalité dominante non prédéterminée par le modèle d'architecture
(3) t : proximité temporelle,
l : complémentarité logique,
c : complétude de la commande,
h : historique et/ou modèle de la tâche (extension de (c) à un contexte plus élaboré),
i : prise en compte de l'incompatibilité possible de modalités,
r : le réseau établi lui même les critères d'intégration.

5.2. Remarques

On distingue deux abords de la multi-modalité dans les systèmes présentés, ce qui rend les comparaisons délicates:
• Certains systèmes présentent une architecture dans le cadre d'une application multi-modale, [Nigay, Bellik, Vigouroux, Bourguet]. Ces systèmes ont l'avantage d'être validés par une expérience pratique, mais il est possible que les concepts inférés présentent une dépendance vis à vis de l'application considérée.
• [Azémard] et [Martin] présentent un modèle d'architecture indépendant à priori d'une application donnée, mais non évalué dans un contexte réel.

Deux types d'application sont proposés permettant de mettre en valeur les apports de la multi-modalité:
• Interrogation de base de données [Nigay, Azémard];
• Construction d'objets graphiques [Bellik, Vigouroux, Bourguet].

Les deux stratégies proposées, dans le contexte de l'interprétation de commandes multimodales, pour intégrer/fusionner des informations exprimées dans des modalités différentes, se heurtent à des obstacles difficiles à surmonter, au moins pour l'instant.

Une fusion tardive, à un niveau d'abstraction élevé, oblige à gérer un nombre éventuellement important d'interprétation concurrentes, ce qui risque de compliquer la sélection de celle(s) que le système retiendra.

Une fusion précoce peut entraîner une multiplication coûteuse des retours arrière (ou défusions).

Une stratégie d'interprétation qui privilégie une modalité (notion de modalité dominante) permet d'éviter en partie ces écueils. Mais des résultats expérimentaux récents [Mignot 93] montrent que, dans un contexte de communication homme-machine autorisant une commande multimodale orale et gestuelle (écran tactile), la modalité dominante, parole ou geste, varie d'un sujet à l'autre et, pour un même sujet, de la réalisation d'une sous-tâche à une autre. Pour être en mesure d'exploiter efficacement la notion de modalité dominante au cours de l'interprétation d'une commande multimodale, il est nécessaire d'approfondir notre connaissance du comportement des futurs usagers de cette forme de communication homme-machine.

D'autre part, la plupart des stratégies d'interprétation proposées par les participants mettent en œuvre une démarche d'analyse essentiellement ascendante (i.e. guidée par les données). Une stratégie flexible (ou opportuniste), ascendante ou descendante en fonction de la robustesse des informations disponibles, permettrait d'utiliser plus efficacement le contexte pragmatique (domaine d'application, objectifs de l'usager, etc…) pour réduire l'ambiguïté des productions multimodales de l'usager et reconstruire ses commandes. Nous songeons, notamment, aux stratégies mises en œuvre en compréhension automatique de la parole continue (cf. [Carbonell 89]).

Toutefois, la structuration et l'interprétation des productions multimodales des usagers exigent de nombreuses études préliminaires. Actuellement on ne dispose d'aucun critère robuste pour regrouper en unités significatives les informations en provenance de l'utilisateur, lorsqu'elles transitent par plusieurs canaux de communication. Par exemple, la proximité temporelle s'avère inefficace si l'on autorise l'utilisateur à combiner librement multimodalités concurrente et synergique.

Des études sont nécessaires pour déterminer si les usagers fournissent spontanément des informations, ou même simplement des indices, susceptibles :

- d'une part, de réduire l'ambiguïté de leurs énoncés multimodaux, c'est-à-dire de faciliter la segmentation du continuum que représente leur production multimodale en termes de commandes (i.e. d'actes de communication élémentaires), surtout dans un contexte de multimodalité synergique,
- et, d'autre part, de faciliter l'identification de leur stratégie d'utilisation de la multimodalité, c'est-à-dire d'aider le système à déterminer s'ils font appel à une multimodalité synergique ou alternée, dans le contexte d'un dialogue multifils (multimodalité concurrente) ou non ?

Si de tels indices existaient et si l'on était capable de les détecter de manière robuste, leur mise en œuvre simplifierait sensiblement l'interprétation d'énoncés multimodaux faisant intervenir, sans contrainte, le geste et la parole.

6. Problèmes et perspectives pour la construction des IHMM [N. Vigouroux, IRIT & Ph. Palanque, LIS]

L'analyse des mécanismes et des stratégies de fusion des énoncés multimodaux en entrée, ainsi que la représentation des connaissances associées à ces mécanismes et stratégies ont été au centre des réflexions de notre atelier.

Elles ont fait apparaître les points suivants relatifs :

• à la complétude des commandes plus ou moins bien résolue par les différentes approches d'intégration des modalités proposées ;
• à la prise en compte de contraintes temporelles sur l'interprétation des commandes de l'opérateur ;
• à l'élaboration d'un modèle d'interprétation d'énoncés multimodaux générique ; ce formalime devrait permettre de spécifier les caractéristiques d'interprétation des énoncés émis par l'utilisateur telles que la coïncidence temporelle, spatiale, opérateur de complétude, …
• à la spécification d'un modèle d'architecture générique indépendant de la modalité et de l'application;

Cette section n'a pas pour objectif de présenter une liste exhaustive des questions et des problèmes en suspens mais de faire émerger une liste potentielle d'axes de recherche autour des thèmes suivants, à savoir :

• les problèmes spécifiques liés à la prise en compte des entrées / sorties dans les IHMM,
• les différents composants intervenant dans une IHMM,
• les modèles d'architecture d'organisation de ces différents composants,
• l'intérêt d'un formalisme générique pour la modélisation de ces différents composants,
• la réutilisabilité et l'extensibilité de ces concepts pour des applications de type travail coopératif (collecticiel multimodal, par exemple).

La section 6.1 est consacrée aux problèmes liés à la prise en compte de la multimodalité en entrée et en sortie, de l'entrelacement des entrées et des sorties dans une perspective d'élaboration d'un modèle générique de prise en compte de la multimodalité dans une IHMM.

La section 6.2 a pour but de montrer le rôle du contrôleur de dialogue et la nécessité d'intégrer à une IHMM des modèles d'utilisateur, de dialogue et d'interaction.

La section suivante (6.3) montre la nécessité de définir une architecture conceptuelle d'organisation de ces différents composants.

La section 6.4 tente de spécifier les caractéristiques d'un formalisme générique qui permettrait de spécifier les critères à utiliser pour interpréter un énoncé émis par l'utilisateur.

Enfin, la dernière section (6.5) aborde le problème de l'évaluation d'une IHMM et du rôle de la multimodalité dans le cadre d'un travail coopératif tel que le collecticiel multimodal.

6.1. Les problèmes inhérents aux énoncés multimodaux dans les IHMM

Pour aborder le problème de la gestion des E/S, nous définissons la communication homme-machine suivant l'axe des entrées utilisateur ⇒ ordinateur et l'axe des sorties (ou retours d'informations) ordinateur ⇒ utilisateur. Dans le cadre d'une communication homme-machine ces deux axes peuvent se chevaucher et générer par conséquent des problèmes spécifiques dus à un entrelacement continu des entrées et des sorties (cas du graphisme animé, de la réalité virtuelle).

6.1.1. Les problèmes inhérents à la fusion des informations multimodales (entrées)

Les interfaces offrant une ou plusieurs modalités à composante langagière (forme écrite) posent des problèmes spécifiques relatifs à l'analyse et à l'interprétation des entrées (résolution des anaphores, des ellipses et des déictiques) d'un énoncé. Cette interprétation nécessite recours à l'utilisation de connaissances linguistiques, d'informations sur l'état du dialogue (à savoir, historique du dialogue, état et historique de la tâche, modèle de l'usager…) pour résoudre les références aux objets, aux concepts, aux références spatio-temporelles…

Dans les interfaces multimodales, l'interprétation de l'énoncé multimodal recourt aux mêmes connaissances (état du dialogue, de la tâche et informations apportées par chaque modalité). Repose-t-elle uniquement sur la complétude de l'énoncé ? Diverses solutions ont été présentées au paragraphe 4 tenant compte essentiellement de caractéristiques temporelles, d'opérateurs de complémentarité et de connaissances sémantico-pragmatiques. Ces solutions tiennent-elles compte de tous les caractéristiques des modalités ?

Le constat suivant peut être fait. En opposition aux IHM à désignation directe où l'énoncé résultant de l'interaction est robuste (i.e. fiable et non ambigue), les IHMM comportant des entrées orales, des systèmes d'analyse du geste, des systèmes de reconnaissance manuscrite d'écriture, … devraient à terme prendre en considération les problèmes dus :

(a) à une mauvaise reconnaissance de l'entrée,

(b) à des entrées non prévues (phénomènes de communication qui peuvent se produire dans des actes de parole ou gestuels tels que hésitation, balbutiement, reprises d'un acte …),

(c) à des informations à considérer comme hors contexte de l'énoncé multimodal.

Sur cette base de constatation, qui n'est bien sûr pas exhaustive, ne conviendrait-il pas de s'intéresser à l'étude de stratégies d'interprétation d'énoncés multimodaux tenant compte :

• de modèles d'erreurs des systèmes d'entrées où la compréhension en l'état actuel des performances des systèmes de reconnaissance de la parole, du geste, de l'écriture cursive, ... est parfois erronée,

• d'une modélisation du comportement de l'usager qui peut résulter de connaissances erronnées de l'usager sur la tâche à réaliser (entrées non prévues telle que les phénomènes d'hésitation, …),

• d'un procédé de rejet d'informations superflues inutiles à l'élaboration de l'énoncé à l'aide de critères de complétude de l'énoncé, …

6.1.2. Génération d'informations multimodales (sorties)

Le problème de la génération de sorties multimodales n'a été qu'évoqué dans le cadre des journées IHM'92. Le premier travail devrait donc consister à définir la notion d'énoncé et d'événements dans le cadre des sorties multimodales. Dans le cas d'un retour, l'énoncé devrait traduire un changement d'état du noyau fonctionnel qui doit être reflété dans l'IHMM. La nature de ces énoncés émanant du noyau fonctionnel reste à définir.

On pourra alors s'interroger sur la nature et le niveau de difficulté des problèmes posés par la gestion des énoncés de retour du noyau fonctionnel vers une IHMM? Quels seraient la stratégie et les critères de ce module de "fission" ?

S'il existait, un tel module aurait-il pour fonction de distribuer les différentes informations à retourner à l'utilisateur en exploitant (en fonction du type de ces informations) le ou les modes les plus adaptés ? Cette stratégie intervient-elle au niveau des différents composants du modèle de Arch (de même que pour le feedback dans les IHM graphiques) ?

Le retour d'information vers l'utilisateur nécessite des études ergonomiques et cognitives pour déterminer le mode le plus adéquat sur lequel elle doit être restituée. Ce mode peut dépendre de l'utilisateur, de l'environnement ou de l'application. Ces études permettraient l'élaboration de critères de "fission" qui associeraient à un type de message (alerte, aide, ordre…) et à son contenu (type de la commande, type des arguments…) le ou la modalité de sortie la plus adéquate.

Nous vous invitons donc à méditer sur un exemple du fonctionnement possible d'un tel module de diffusion qui pourrait être le pendant du "mets ça là" au niveau des sorties :

Question utilisateur : Comment détruire ce cercle rouge ?
Une réponse multimodale que pourrait fournir l'application serait :
• mode vocal : Mets le ici ?
• mode désignation directe : le pointeur de la souris désigne la poubelle.

6.1.3. Entrelacement des entrées/sorties continues

Le problème de l'entrelacement des E/S a été abordé en particulier dans [Baudel 91] ainsi que dans:
• dans le cadre de l'interface multimodale du modeleur (cf. paragraphe 4.3) ;
• dans le cadre de VoicePaint [Gourdol 91] (dessin en manipulation directe et intervention vocale pendant le dessin).

Ce problème d'entrelacement des E/S semble très lié à deux autres problèmes : ceux de la granularité et du caractère discret ou continu des informations multimodales. Dans le cas d'informations discrètes, si on réduit la granularité des informations on peut considérer l'entrelacement des E/S comme un basculement rapide entre entrées puis sorties d'informations multimodales. On se ramène ici aux deux situations précédentes (entrées multimodales et sorties multimodales gérées de manière indépendante).

Dans le cas d'informations continues il est nécessaire de gérer les problèmes d'exploitation simultanée des modalités à la fois en entrée et en sortie (i.e. l'étude des relations entre les entrées et les sorties). Ce travail nécessitera tout d'abord d'identifier les interactions à caractère continu et de s'interroger sur le "droit" qu'a l'utilisateur d'interrompre la tâche en cours. Ceci soulève le problème de la prise en compte d'actions rapides de la part de l'utilisateur au sein même de l'interface.

6.2. Le contrôleur de dialogue d'une IHMM

Il est clair que pour tout le groupe de travail que le module dialogue à sa place dans une architecture d'interface homme-machine. Cependant ce terme ne représente pas la même chose pour nous tous :

• pour les concepteurs d'interfaces graphiques, ce module se réduit très souvent à la modélisation de l'interaction 2D directe d'objets;

• pour les chercheurs de la communauté scientifique en langage naturel - écrit ou oral - , il englobe tous les présupposés des actes de langage du dialogue homme-homme.

En effet, le dialogue en conversation humaine est soumis à des lois et obéit à des principes [Caelen, 1992] de coopération, de négociations, de lois socio-culturelles, … . Avec une machine, les fondements du dialogue sont plus difficiles à discerner : que cherche à faire un utilisateur ? construire une scène de synthèse d'image, consulter une base hypermédia, donner des ordres à un robot ?

Ces points de vues nous amènent à définir la nature et la place même du contrôleur de dialogue homme-machine dans une interface IHMM et à nous interroger sur les conséquences des libertés d'expression laissées à l'interlocuteur en interaction multimodale pour le contrôleur de dialogue.

Ci-dessous nous allons évoquer succinctement le rôle du modèle de dialogue, du modèle de l'interaction, du modèle de l'utilisateur et du problème du dialogue à plusieurs fils d'activité dans une IHMM.

6.2.1. Le modèle de dialogue

On peut s'interroger sur la nécessité dans le cadre d'un mécanisme d'intégration de la multimodalité de disposer de connaissances sur l'état du dialogue à savoir :

• des stratégies de dialogue en relation avec l'état de la tâche;
• d'un modèle d'usage des modalités en relation avec la tâche;
• d'informations contextuelles mises à jour au fur et à mesure de l'interaction multimodale et du déroulement de la tâche.

6.2.2. Le modèle de l'interaction

L'ensemble des interactions offertes à un instant donné à l'utilisateur est facile à représenter dans les IHM (la technique généralement utilisée consiste à faire apparaître en "grisé" les interactions non disponibles).

Dans le cas d'une IHMM, en plus des interactions graphiques il faut expliciter dans ce modèle :
• les modes de communication disponibles (vocales ou autres),
• l'ensemble des entrées que peut faire l'utilisateur avec chacun de ces modes (comment faire ceci ? liste des commandes vocales affichées ou consultables...),
• et les types de multimodalité possibles.

Ceci se ramène-t-il au problème de l'aide contextuelle ? Dans le cas où certaines commandes ne sont pas accessibles, il faut que l'interface soit en mesure d'expliquer pourquoi la commande est inactive ainsi que comment la rendre active (i.e. rendre explicite à un moment donné les modalités et combinaisons offertes).

6.2.3. Le modèle de l'utilisateur

La question se pose de savoir intégrer une modélisation de l'utilisateur. Cette modélisation doit-elle être explicite ou répartie dans les différents couches de l'interface ?

6.2.4. Le dialogue à plusieurs fils d'activités dans une IHMM

Le dialogue à plusieurs fils d'activités soulèvent :
• la gestion du focus, plus particulièrement le basculement du focus (basculement d'un énoncé à un autre, silmutanéité dans la formulation d'énoncés indépendants),
• et la gestion des informations partagées par les tâches qui se déroulent en philigramme dans chaque fils d'activité.
Les environnements graphiques tels que Windows 3.1. disposent de mécanismes de base qui offrent la possibilité de partager des connaissances entre les différentes activités. Cependant aucune boîte à outils ne rend compte dans un fil d'activité de l'effet de l'action réalisée dans un autre (aspect présentation).

Ces dialogues à multifils d'activité véhiculent des recommandations et des contraintes qu'il faudra considérer dans les modèles conceptuels des nouvelles architectures.

6.3. Problèmes et proposition d'un modèle d'architecture conceptuel d'IHMM

Le modèle Arch a été utilisé comme cadre conceptuel d'une IHMM pour présenter les différents niveaux d'abstraction auxquels peut se situer le mécanisme de fusion pour la représentation et la prise en compte des informations modales en entrée (Voir Paragraphe 3). Toutefois, le modèle Arch ne rend pas compte de tous les niveaux de représentation de connaissance nécessaires au module d'interprétation. Parmi ces connaissances on peut remarquer l'absence des modèles utilisateur, de dialogue et d'interaction. En effet, dans les interfaces graphiques classiques de tels modèles ne sont pas indispensables au bon fonctionnement de l'application même si leur prise en compte dans le fonctionnement de l'interface permet d'accroître de façon significative l'efficacité du couple Homme-Machine.

Dans le domaine des IHMM, l'intégration de tels modèles est indispensable au module de fusion. En particulier elle permet de résoudre les problèmes d'incomplétude ou d'ambiguïté de la commande et de bruit (événements à rejeter qui sont hors contexte ou incompréhensibles (cf 6.1), dans le cas où les informations présentes sur les autres modes sont insuffisantes pour résoudre tous ces problèmes. L'intégration d'un modèle de dialogue peut permettre par exemple de guider le dialogue en vue d'une requête d'informations vers l'utilisateur (cf. section 6.2).

Il faudrait donc définir une architecture conceptuelle pour les application proposant à l'utilisateur une IHMM. Une telle architecture devrait :
• expliciter les différents modules qu'elle intègre ainsi que leur type (le type des informations qu'ils contiennent). Pour l'instant, il semblerait qu'elle doive contenir au moins les modules de Arch, un module de fusion, un modèle de l'utilisateur, un modèle de l'interaction ainsi qu'un modèle de la tâche ;
• expliciter les relations qu'entretiennent ces différents modules (utilisation des données et/ou des services d'un module par un autre, acheminement d'informations vers un autre module, …) ;
• expliciter les différents niveaux d'abstraction entre les modules.

6.4. Généricité ou pas dans les interfaces multimodales ?

Est-il réaliste de parler d'un formalisme générique d'interface ? Ce terme de généricité s'applique-t-il :
• aux modalités d'entrées/sorties de l'interface,
• aux applications à envisager,
• à un mécanisme générique d'interprétation de l'énoncé multimodal,
• …

Le problème de la générité d'une IHMM peut être abordé sur les cinq modèles de représentation des connaissances d'une interface multimodale à savoir [Caelen 91] :

(a) le niveau de la représentation de la tâche : par exemple, objets, modèles des règles, …
(b) le niveau de la modélisation de l'interaction qui décrit l'ensemble des enchaînements possibles de l'interaction (Réseau de Petri, scripts, grammaires, Augmented Transition Network, …),
(c) le niveau de modèles du dialogue (système à base de règles, Augmented Transition NetWork, …),
(d) le niveau de modélisation de l'utilisateur (état de croyance de l'usager sur la tâche, …),
(e) et les modèles de langage qui décrivent les langage utilisés (langage naturel, gestuel, …).

Généralement dans les interfaces homme-machine à manipulation directe les modèles c) et d) ne servent qu'à améliorer l'efficacité de l'interface homme-machine et le modèle e) est inutile. Dans le cas des IHMM ces modèles sont utiles et accélèrent le module d'interprétation (vs, la fission) de l'énoncé (vs message) multimodal.

Généralement ils sont modélisés par des formalismes différents ce qui nécessite la mise en œuvre de techniques de correspondance si l'on envisageait leur utilisation conjointe dans une IHMM. Une autre orientation qui semblerait préférable consisterait à définir un formalisme générique permettant la description de ces différents modèles.

Notre propos sera moins ambitieux en ce sens, que nous nous limiterons à donner quelques directions pour lesquelles il nous semble indispensable d'aborder le problème de la généricité :

• formalisme générique de description de plusieurs types de tâches;
• formalisme générique pour la prise en compte d'un nombre quelconque de modalités dans une stratégie d'interprétation d'énoncé multimodal. On pourra se reporter au Rapport de Synthèse du Groupe Modalités avancées dans ces mêmes actes.

6.5. L'évaluation d'une IHMM

Le développement d'IHMM opérationnelles autorisera l'observation du comportement du sujet en situation naturelle d'interaction multimodale. Dans ce cadre, il conviendrait d'étudier l'intérêt de l'utilisation d'une IHMM et notamment :

- Impact sur la charge cognitive de l'usager,
- Impact sur le taux d'erreur de manipulation,
- Conséquences sur le temps de réaction de l'application,
- Accroissement du domaine des applications traitables,
- …

Tous ces perspectives peuvent êtres étudiées dans des situations d'interaction multimodale pour des tâches de type travail coopératif. Dans des tâches de type coopératives , vont se poser de nouveaux problèmes tels que :

• identification de l'interlocteur (machine ou homme et de l'intention de l'utilisateur),
• modèles d'architectures conceptuels pour des IHMM

7. Références

[Azémard 92a]
F. Azémard, U. Arronategui, T. Huguet, J.P. Macchion, F. Mieulet : Interprétation et gestion du dialogue sous LOIR. Séminaire Dialogue, Pôle Parole et Langage Naturel du GDR-PRC CHM, Dourdan (Avr. 1992).

[Azémard 92b]
F. Azémard, J.P. Macchion, F. Mieulet: Multimodal : Le réflexe pour une stratégie d'interprétation dans Actes IHM'92, Telecom Paris (déc. 1992)

[Baudel 91]
T. Baudel. Spécificités de l'interaction gestuelle dans un environnement multi-modal. IHM91, dans Actes d'IHM'91.

[Baudel 93]
T. Baudel, A. Braffort. Reconnaissance de gestes de la main en environnement réel. Actes de la conférence Informatique'93, L'interface des Mondes Réels et Virtuels. Montpellier, Mars 1993, pp. 207-208.

[Bellik 92a]
Y. Bellik, D. Teil : Les types de multimodalité dans IHM'92, Telecom Paris (déc. 1992) pp. 22-28.

[Bellik 92b]
Y. Bellik, D. Teil : Définitions terminologiques pour la communication multimodale, dans IHM'92, Annexes, pp. 229-232.

[Béroule 91]
D. Béroule : The ADAM model : an actual present tool for vocal human-computer communication. Taylor M.M., Néel F. Bouwhuis D.G in Proceedings of the Second Venaco Workshop onthe Structure of Multimodal Dialogue, September, 199, pp.

[Béroule 92]
Béroule, D. (1992). "Représentation Topologique Spatio-Temporelle pour la Communication Homme-Machine", Cognition, Perception, et Action dans la Communication Parlée, J.Caelen, J-L.Schwartz, C.Abry (Eds), dans Presses Universitaires de France, Paris.

[Bolt 80]
R. Bolt."Put-That-There": Voice and Gesture at the Graphics Interface. Computer Graphics, 14(3), July 1980, pp 262-270, Proc. ACM SIGGRAPH, 1980.

[Bourguet 92a]
M.L. Bourguet, J. Caelen : Interfaces Homme-Machine multimodales : gestion des événements et représentation des connaissances" dans ERGO-IA'92, pp. 124-134.

[Bourguet 92b]
M.L. Bourguet M.-L. (1992) ICPplan : dialogue multimodal pour la conception de plans architecturaux. 19èmes JEP, Bruxelles, mai, pp. 369-374.

[Bourguet 92c]
M.-L. (1992) Conception et réalisation d'une interface de dialogue personne-machine multimodale. Thèse de l'Institut de la Communication Parlée de Grenoble, Mention Signal Image Parole, Novembre.

[Caelen 92]
J. Caelen : Compte-Rendu du "Workshop" IHM-M organisé par le GDR-PRC "Communication Homme-Machine" à Dourdan les 13 & 14 avril 1992, dans Actes d'IHM'92, Annexes, pp. 213-228.

[Capdevielle 93]
Capdevielle O. : Modalités Avancées, Rapport de synthèse de l'atelier Modalités Avancées à IHM'92.

[Coutaz 87]
Coutaz J. : PAC: an Implementation Model for Dialog Design. In Proc. Interact'87, (Stuttgart, Sept. 1987), H-J. Bullinger, B. Shackel ed., North Holland, pp. 431-436.

[Duby 93]
Duby J.C., David B. : Collecticiel, Rapport de synthèse de l'atelier Collecticiel IHM'92.

[Gaildrat 83]
V. Gaildrat, R. Caubet, F. Rubio, Conception d'un modeleur déclaratif de scènes tridimensionnelles pour la synthèse d'images", MICAD'93, Paris, 1993.

[Gourdol 91]
A. Gourdol : Architecture des Interfaces Homme-Machines multimodales, mémoire de DEA Informatique, IMAG, Grenoble, Juin, 1991.
[Martin 92]
J.-C.Martin, D. Béroule, : Eléments de modélisation connexionniste pour l'interaction homme-machine multimodale.IHM92, dans Actes d'IHM'92, pp. 43-48.
[Martin 93]
J.-C. Martin : Interaction Homme-Machine Multimodale et Neurobiologie de la Perception. Colloque Images et Langages : Multimodalité et Modélisation cognitive, Paris, 1-2 Avril 1993, à paraître.
[Lunati 91]
Lunati J.M. and Rudnicky A.I. Spoken Language Interfaces: The OM system. In Proc. CHI'91 Human Factors on Computing Systems (News Orleans, April 27-May 2 1991), ACM Press, pp. 453-454.
[Nigay 91]
L. Nigay, J. Coutaz : Building User Interfaces: Organizing Software Agents. Proc. ESPRIT'91 Conference (Bruxelles, Nov. 1991), pp. 707-719.
[Nigay 92]
L. Nigay, J. Coutaz : Interfaces multimodales et architecture : fusion et parallélisme, dans Actes IHM'92, Telecom Paris (déc. 1992) pp. 29-35.
[Pfaff 85]
G. Pfaff, P.J.W. Ten Hagen : Seeheim workshop on User Interface Management Systems. Berlin : Springer-Verlag.
[Bellik 92]
Y.Bellik, D. Teil : Multimodal Dialogue Interface, in WWDU'92 Berlin 1-4 Sept 92.
[Salisbury 90]
M.W. Salisbury, J.H. Hendrickson, T.L. Lammers, C. Fu, S.A. Moody : Talk and Draw : Bundling Speech and Graphics, in IEEEComputer, 23(8), August, 1990, pp. 59-65.
[Sowa 84]
J.F. Sowa : Conceptual Structures, information processing in mind and machine. Addison Wesley, (1984).
[Teil 91]
D. Teil, Y. Bellik : Multimodal dialogue interface on a PC-like workstation, in proceedings of 2nd Venaco Workshop ESCA ETRW, Maratea, September, 1991.
[UIMS 92]
The UIMS Tool Developers Workshop : A Metamodel for the Runtime Architecture of an Interactive System. In SIGCHI bulletin, ACM Press, January, 1992, 24 (1), pp.32-37.
[Vigouroux 92]
N. Vigouroux, V. Gaildrat, R. Caubet, G. Pérennou : Une architecture d'interface pour l'interprétation d'informations multimodales : application à un modeleur déclaratif de scènes dans Actes d'IHM'92, Telecom Paris (déc. 1992) pp. 49-56.