

MATIS

un système multimodal d'information sur les transports aériens

**Pôle Interfaces Multimodales du PRC Communication Homme-
Machine**

Laurence Nigay

**Laboratoire de Génie Informatique-IMAG
BP 53 , 38041 Grenoble Cedex 9
Tel. 76 51 44 40, 76 51 48 54
E-mail : Laurence.Nigay@imag.fr, Joelle.Coutaz@imag.fr**

1. Présentation de MATIS

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 :

- manipule la souris et spécifie sa requête par manipulation directe.
- saisit au clavier et spécifie sa requête en langue naturelle écrite ou en remplissant un formulaire.
- utilise le microphone et dicte sa requête. Par exemple : <I would like to know the Usair flights from Pittsburgh to Boston>. Le système de reconnaissance de la parole utilisé est Sphinx [Lunati 91], un système qui assure une reconnaissance multilocuteur de la parole continue.
- choisit 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 est un système multimodal de type synergique dans la classification introduite dans [Nigay 93]. De plus il 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.

Dans cet article nous présentons tout d'abord l'interface en sortie et en entrée de MATIS. Ces deux paragraphes permettent de fixer les possibilités offertes par le système. Le développement de MATIS a permis une étude sur l'architecture logicielle des systèmes multimodaux ainsi que sur l'intégration des modalités (mécanisme de fusion des événements, co-références multimodales). Cette étude fait l'objet des paragraphes 3 et 4.

2. Interface en sortie de MATIS

MATIS a été développé sous l'environnement NeXTStep avec le générateur d'interfaces Interface Builder ; MATIS s'exécute donc sur machine NeXT.

A la figure 1, nous présentons une copie d'écran de MATIS où nous retrouvons le standard de présentation NeXTStep des applications NeXT. Dans la description de cette interface, certaines caractéristiques sont héritées de NeXTStep, nous les marquons d'une astérisque (*). Les autres héritées du système SPHINX, sont marquées par (@).

L'interface en sortie se compose de plusieurs fenêtres graphiques :

- La colonne d'icônes, à droite de l'écran contient les applications du bureau électronique NeXTStep (*). Elle ne fait donc pas partie de MATIS.
- Au coin supérieur gauche de l'écran, le menu principal de MATIS offre cinq options. En particulier, l'option "Tools" donne accès à quatre fenêtres complémentaires intitulées : "Request tools", "Requests History", "Notepad" et "Speech panel".
- La fenêtre "Requests Tools" en haut et au milieu de l'écran, offre un ensemble d'outils de spécification de requête. Chaque outil est représenté par une icône. La sélection de l'une de ces icônes fait apparaître un formulaire permettant de spécifier un champ d'une requête. Par exemple, la sélection de l'icône "City", donne accès à la liste des villes disponibles. Ces outils permettent de spécifier une requête par manipulation directe.
- La fenêtre "Requests History" (en bas à gauche de l'écran) affiche les requêtes déjà envoyées à la base de données. Par les opérations copier/coller, il est possible de réutiliser une requête ou une partie d'une requête. Ainsi l'utilisateur a la possibilité de soumettre à nouveau une requête.
- La fenêtre "Notepad" (en bas à droite de l'écran), permet de centraliser des informations intéressantes sur des vols et de rajouter des annotations.
- La fenêtre "Speech Panel" (non contenue par la figure 1) présente le résultat de l'analyse syntaxique de la dernière phrase en langage naturel. C'est une information utile à la mise au point du système.
- La fenêtre "Office Manager" (en haut et au milieu de l'écran) met en évidence les applications connectées (c.-à-d., MATIS) au système de reconnaissance de la parole (@). Cette fenêtre affiche aussi la dernière phrase reconnue par le système de reconnaissance de la parole, ou les messages d'erreur en cas d'échec de la reconnaissance. C'est aussi la zone d'édition des phrases écrites en langage naturel. A la figure 1, la dernière phrase reconnue est "USAIR FLIGHT FROM PITTSBURGH TO BOSTON SERVING A MEAL".
- En haut à droite de l'écran, une icône traduit l'état courant du système de reconnaissance de la parole. L'état est soit prêt, soit en écoute, soit en cours de recherche. A la figure 1, l'icône représente un microphone et le système est à l'écoute (@).
- La fenêtre "Request" est utilisée pour spécifier une requête. Elle contient deux parties : les deux boutons en haut du formulaire et les champs constituant une requête.

Les icônes des deux boutons sont inspirées de l'environnement NeXTStep. Lorsque la requête est vide, le livre du premier bouton est fermée et le bouton est inactif (en grisé). La requête ne peut être envoyée à la base de données. Lorsque la requête contient assez d'informations (c.-à-d., la ville de départ ou d'arrivée), le livre est ouvert et le bouton est actif. Sa sélection provoque l'envoi de la requête à la base de données. Le deuxième bouton affichant le symbole du "recyclage" est toujours actif et permet de mettre à blanc tous les champs de la requête.

Comme les champs d'une requête sont nombreux, nous les avons divisés en deux catégories selon leurs fréquences d'utilisation. Les champs obligatoires ou très utilisés sont affichés par défaut lors de l'ouverture d'une nouvelle requête (c.-à-d., From, To, Dep. Time, Arr. Time). L'utilisateur peut ajouter d'autres champs dans le formulaire par manipulation directe en utilisant les outils de la fenêtre "Requests Tools" ou en les spécifiant en langage naturel (écrit ou oral). Ainsi, comme le montre la figure 1, l'utilisateur a énoncé "USAIR FLIGHT FROM PITTSBURGH TO BOSTON SERVING A MEAL". La phrase reconnue a provoqué l'ajout automatique de deux champs intitulés "Airline" et "Meal".

Une requête vide est affichée par défaut au lancement de MATIS. MATIS offre un dialogue à plusieurs fils d'activité : il est possible de commencer une nouvelle requête sans achever la requête courante. Cette possibilité est illustrée à la figure 1 où deux requêtes sont en cours de spécification. Pour rendre active une requête, il suffit de sélectionner la fenêtre correspondante. Celle-ci est alors affichée au premier plan et définit le nouveau contexte de l'interaction.

- La fenêtre "Results of request", au centre de l'écran, affiche sous forme de table le résultat d'une requête soumise à la base de données. Les informations contenues dans la table peuvent être sélectionnées et copiées dans une autre fenêtre, par exemple la fenêtre "Notepad".

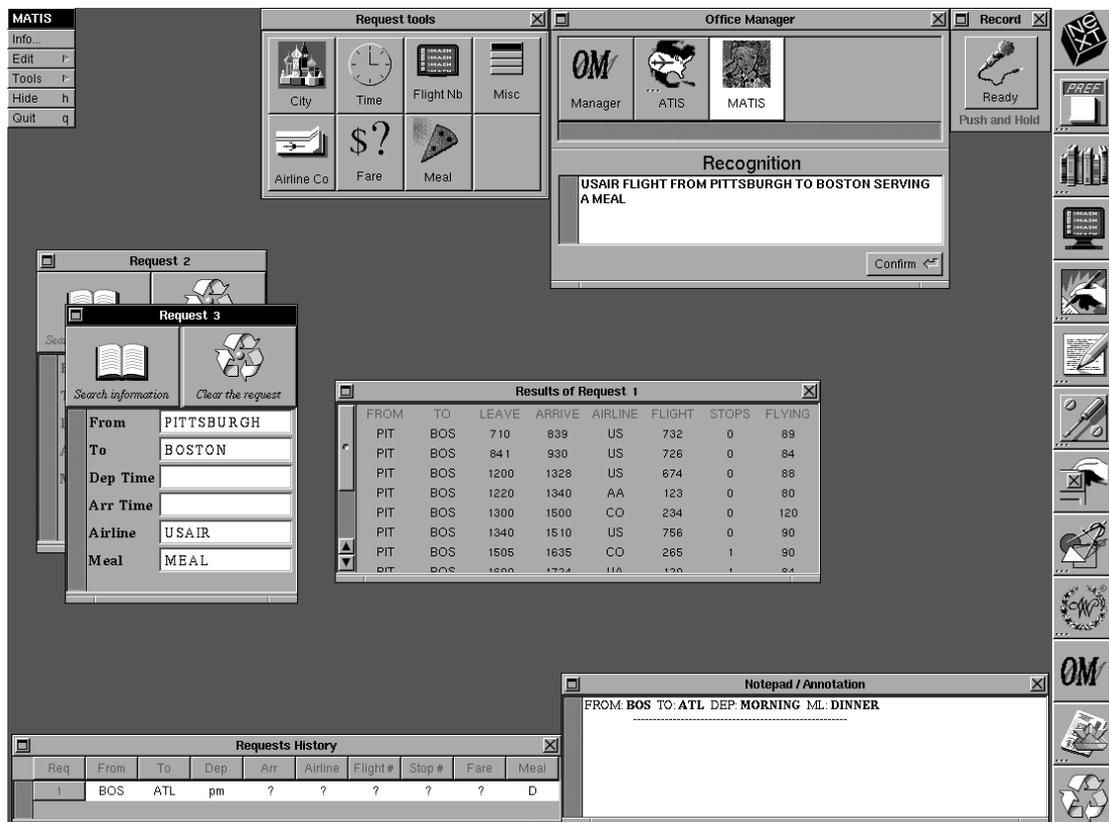


Figure 1 : copie d'écran du système MATIS.

2. Interface en entrée de MATIS

L'utilisateur de MATIS exprime des requêtes afin de planifier un voyage. Pour cela, l'interface en entrée de MATIS offre les deux métaphores introduites par D. Norman [Norman 86] :

- la métaphore conversationnelle,
- et la métaphore du monde réel.

Dans le cas de la métaphore conversationnelle, l'utilisateur a le choix entre :

- le quasi langage naturel oral ou écrit , comme "I would like to know the flights from Pittsburgh to Boston serving a meal",
- ou le langage artificiel écrit directement dans les champs d'une requête, comme "PIT" qui est le code de la ville Pittsburgh.

D'autre part, l'utilisateur peut spécifier une requête par manipulation directe avec la souris (métaphore du monde réel). Chaque champ d'une requête fait l'objet d'un formulaire accessible par la fenêtre "Requests tools" ; un formulaire permet la spécification par manipulation directe d'une valeur d'un champ particulier d'une requête.

Enfin l'utilisateur peut combiner les langages offerts et la manipulation directe. Par exemple l'utilisateur énonce la phrase "Flights from Pittsburgh to this city" tout en sélectionnant avec la souris la ville de Boston dans la liste des villes ; il peut alors préciser sa demande en saisissant au clavier "1500" dans le champ de l'heure de départ de la requête.

L'interface en entrée de MATIS a été spécifiée de façon formelle en utilisant le langage UAN (User Action Notation [Hartson 92]) dans [Coutaz 93].

3. Le 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 [UIMS 92] dont il organise le composant Contrôleur de Dialogue en une hiérarchie d'agents PAC [Coutaz 87]. La figure 2 présente l'architecture logicielle de MATIS.

3.1. Les composants

L'utilisateur et le Noyau Fonctionnel produisent et consomment des informations par l'intermédiaire du Contrôleur de Dialogue, le composant clé de voûte de l'architecture.

Le Noyau Fonctionnel correspond à la base de données contenant les informations sur les transports aériens. L'accès à ces informations se fait par l'intermédiaire de requêtes en langage S.Q.L. (System Query Language).

Le composant Adaptateur de Domaine a un rôle de traducteur. Il traduit le résultat des requêtes soumises à la base de données sous une forme textuelle compréhensible par le Contrôleur de Dialogue. Ce dernier en produit une présentation graphique perceptible par l'utilisateur. Symétriquement, l'Adaptateur de Domaine traduit les requêtes exprimées dans le formalisme du Contrôleur de Dialogue en une requête S.Q.L. Ce composant sert donc d'interface conçue pour absorber les changements entre ses

voisins directs. Par exemple, le changement de la base de données provoquerait uniquement une modification du composant Adaptateur de Domaine.

Le Contrôleur de Dialogue est organisé en une hiérarchie d'agents PAC qui forme un pont entre l'Adaptateur de Domaine et les Techniques de Présentation. Avant de détailler la hiérarchie d'agents au paragraphe 3.2, nous devons préciser les responsabilités de ce composant : enchaînement des tâches, transformation de formalisme et mécanisme de correspondance.

- Une tâche correspond à un objectif de l'utilisateur et définit un ou plusieurs fils de dialogue, reflète sa structuration. L'enchaînement entre les fils n'est pas fixe et dépend du plan opportuniste de l'utilisateur. Cette observation conduit à exploiter l'approche multi-agent en associant un ensemble d'agents à un fil de dialogue.
- Par fonction, l'Adaptateur de Domaine et les Techniques de Présentation reposent sur des formalismes distincts, le premier dirigé par des considérations opératoires, le second par des techniques de présentation comme celles offertes par une boîte à outils. Les informations en transit dans le Contrôleur de Dialogue sont donc transformées.
- Les changements d'état dans l'Adaptateur de Domaine doivent être reflétés dans la présentation et vice-versa. Aussi le Contrôleur de Dialogue maintient des liens entre les objets conceptuels et les objets de présentation.

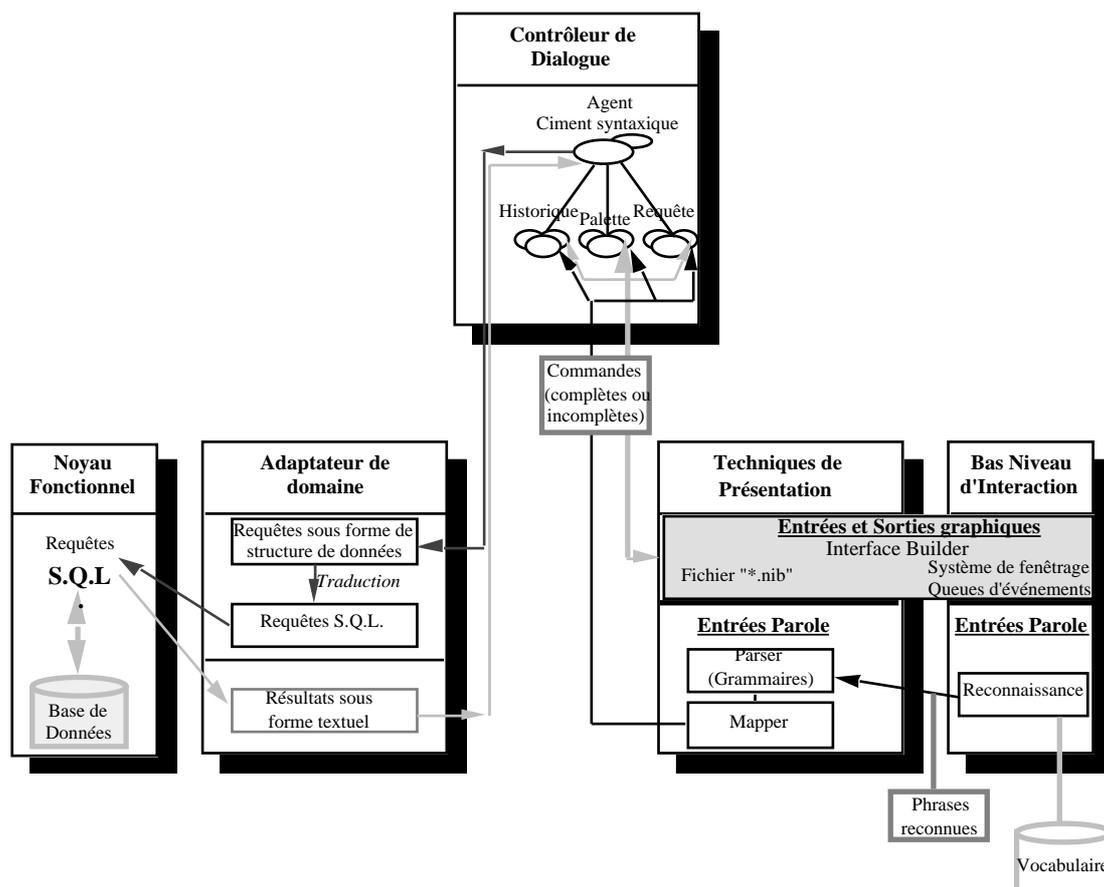


Figure 2 : architecture logicielle du système MATIS.

Les Techniques de Présentation définit les règles de correspondance entre objets de présentation et objets d'interaction. En sens inverse, ce composant abstrait les informations spécifiées par l'utilisateur dans le formalisme du Contrôleur de Dialogue. Ce composant constitue le plus haut niveau d'abstraction où intervient la notion de modalité. Dans le cas de l'implémentation de MATIS, ce composant est scindé en deux parties : l'une est dédiée à la présentation graphique et l'interaction par manipulation

directe sur les objets graphiques (partie englobée par Interface Builder), l'autre est dédiée à l'analyse des phrases en langage naturel (écrites ou parlées).

Le composant Bas Niveau d'Interaction désigne la plate-forme d'accueil logicielle et matérielle. Ce niveau regroupe les services d'acquisition, d'estampille et de répartition des événements mais aussi les objets d'interaction des boîtes à outils et les systèmes de reconnaissance spécialisés. Par exemple, un message vocal est transmis au composant Techniques de Présentation sous forme d'un événement estampillé dont le contenu est une chaîne de caractères.

3.2. Le Contrôleur de Dialogue organisé en une hiérarchie d'agents PAC

La hiérarchie d'agents composant le Contrôleur de Dialogue est présentée à la figure 3. On trouvera dans [Nigay 91] un ensemble de règles heuristiques permettant d'identifier ces agents PAC en fonction de l'interface à gérer.

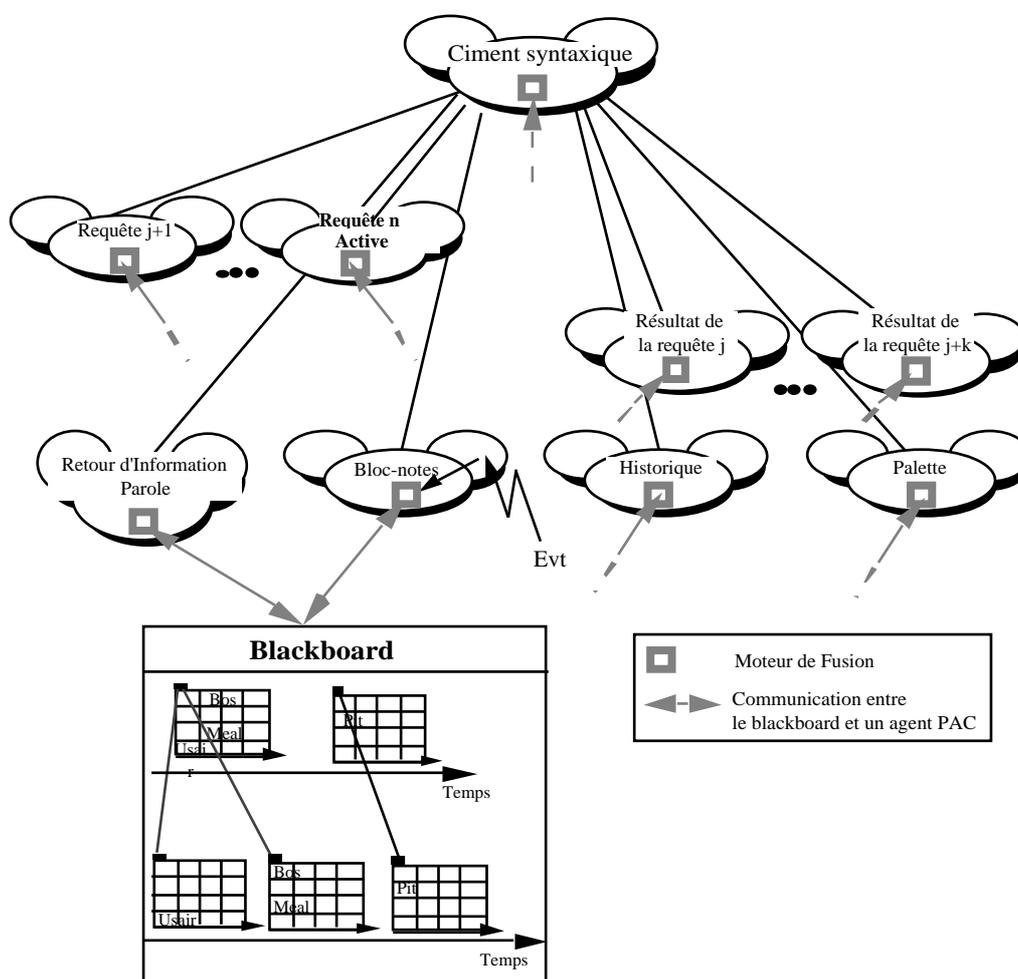


Figure 3 : hiérarchie d'agents PAC composant le Contrôleur de Dialogue.

La hiérarchie contient 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 (fenêtre intitulée "Results of request").

- 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) ; cette information est affichée dans la fenêtre "Speech Panel".
- L'agent *Bloc-notes* gère un bloc-notes où l'utilisateur peut noter des informations relatives à son plan de voyage (fenêtre intitulée "Notepad").
- L'agent *Historique* gère l'historique des requêtes émises (cet historique peut être édité dans la fenêtre "Requests History").
- L'agent *Palette* correspond à la palette d'outils offerts à l'utilisateur pour spécifier une requête (fenêtre intitulée "Requests Tools") comme 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. L'intégration des modalités

Comme le montrent les exemples d'énoncés de requêtes décrits en introduction, MATIS est un système multimodal de type synergique. Il permet cependant la spécification en parallèle de deux requêtes indépendantes. Au niveau de la tâche il est donc aussi de type concurrent. Rappelons aussi que 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.

L'intégration des modalités se traduit par le traitement en parallèle et la fusion des informations en entrée, spécifiées selon différentes modalités. Nous présentons ici le moteur de fusion intégré dans le système MATIS.

Le stratégie 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. Donc la complémentarité structurelle est 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 s'appuie sur le temps et combine des événements ayant des intervalles de temps qui s'entrelacent (parallélisme ou pseudo parallélisme).
- La Macro Fusion s'appuie 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

Le moteur de fusion manipule des informations stockées dans un *blackboard*. Une structure de représentation commune unique, appelée creuset, est utilisée. Comme le montre la figure 4, un creuset est un objet structuré à deux dimensions. L'un des axes définit les composants structurels de l'information (par exemple les éléments d'une commande). L'axe des temps identifie l'ordre de remplissage des composants structurels. La présence de plusieurs valeurs pour un même composant peut révéler une redondance ou des incohérences. Le creuset doit être rempli à l'intérieur d'une fenêtre temporelle, sinon il est abandonné.

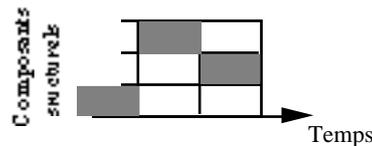


Figure 4 : L'objet creuset comme structure de représentation commune pour la fusion.

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 dé-fusionner facilement, les informations étant toutes exprimées dans le même formalisme. Un creuset correspond à un ensemble de cases. La sémantique de l'information contenue dans chaque case d'un creuset est fixée. Dans le cas de MATIS, la case numérotée 1 correspond à la ville de départ. Dans une autre application, la case numérotée 1 permettra de stocker un paramètre d'une commande (un objet graphique par exemple). Cependant le moteur de fusion manipule une case sans connaître la sémantique de l'information stockée, s'appuyant uniquement sur la structure globale du creuset (case vide ou remplie). Ainsi du point de vue du moteur de fusion, cette structure de représentation, appelée creuset, est indépendante du domaine de l'application.

Le moteur de fusion, de part ses critères d'intégration et manipulant des creusets, est donc indépendant du domaine de l'application.

5. Exemples

Nous trouverons dans [Nigay 93] l'exemple de référence de fusion "mets ça là" qui illustre notre mécanisme. À l'origine, trois creusets sont créés, l'un correspondant à l'événement parole "mets ça là", 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.

Nous développons ici un exemple d'énoncés accepté par le système MATIS. Les figures 5.1 et 5.2 présentent les étapes de fusion et de dé-fusion de cet exemple. Ce processus est effectué au niveau du Contrôleur de Dialogue et est déclenché par des agents PAC distincts. L'exemple est le 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 puisqu'ils appartiennent à une même fenêtre temporelle (Macro Fusion). L'utilisateur prononce ensuite la phrase <Two flights from > tout en sélectionnant la ville <Pittsburgh>. La sélection est traitée en première (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 <Two Flights from> est disponible, il est alors nécessaire

de dé-fusionner la sélection <Pittsburgh>, qui peut alors être fusionnée avec l'événement Parole, émis en parallèle de la sélection (Micro Fusion).

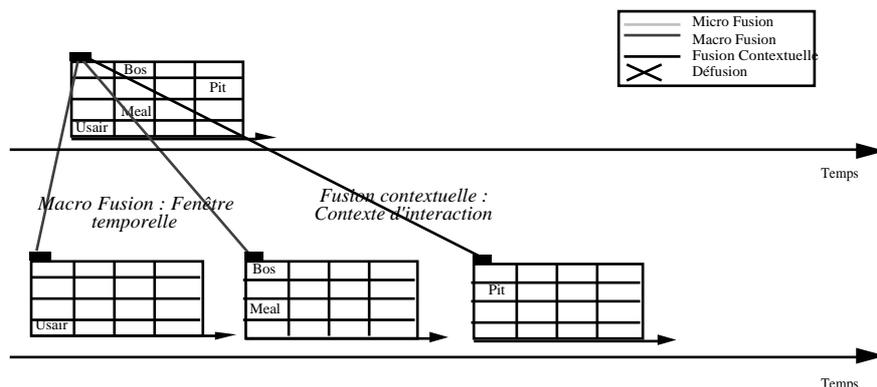


Figure 5-1 : 1ère étape : Macro Fusion puis Fusion Contextuelle.

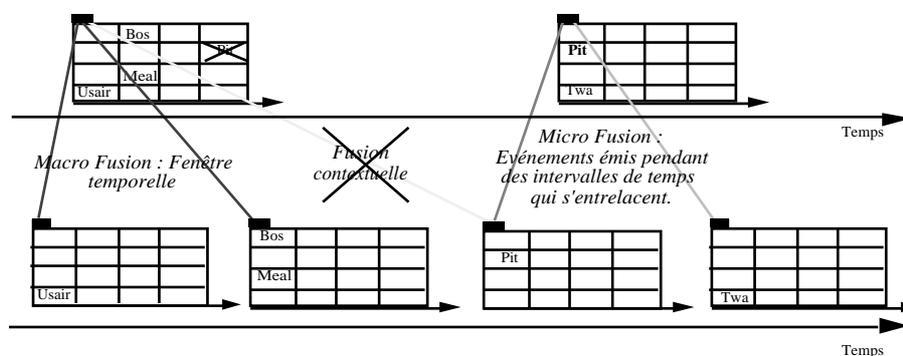


Figure 5-2 : 2ème étape : Dé-fusion et Micro Fusion.

Dans notre exemple, des agents distincts reçoivent les informations à fusionner à chaque étape. Ces agents fournissent avant même que la commande soit complète des retours d'informations spécifiques au niveau d'abstraction. Par exemple l'agent qui reçoit la sélection Pittsburgh produit un retour d'information immédiat traduisant la sélection, en grisant la case sélectionnée. Comme le montrent les figures 5, chaque information est rangée dans les cases structurales idoines d'un creuset. Le creuset résultant de la Macro Fusion contextuelle et de la Fusion Contextuelle, de la figure 5-1, est reçu par un agent, père des agents ayant reçu les messages contenant l'information originale. Lors de la deuxième étape, figure 5-2, la Fusion Contextuelle est défaite et le creuset contenant l'information Pittsburgh est fusionné avec le nouveau creuset.

6. Conclusion

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

Nous avons montré que notre modèle d'architecture PAC-Amodeus fournit un cadre de réalisation pour le traitement en parallèle et la fusion de données en entrée. La fusion des données est effectuée au niveau du composant Contrôleur du Dialogue et s'appuie sur la hiérarchie d'agents PAC qui organise ce composant. Le moteur de fusion manipule des informations représentées dans un formalisme unique, le creuset. 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 domaine 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 dé-fusionner. 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.

Notre modèle offre aussi un support intéressant pour fournir un retour d'information immédiat. En effet un retour d'information peut être généré par chaque agent dans la hiérarchie. Ainsi il est possible de définir des retours d'information partiels et spécifiques au niveau d'abstraction, avant même que le mécanisme de fusion soit fini, pour une commande donnée.

PAC-Amodeus offre aussi une grande flexibilité. Par exemple, il est possible de changer ou de rajouter une modalité d'interaction en ne modifiant que les composants Niveau Bas d'interaction et Techniques de Présentation, le Contrôleur de Dialogue étant indépendant des modalités.

Notre travail futur s'oriente sur l'intégration d'un réel modèle du dialogue au sein de l'architecture logicielle et l'exploitation du contexte de l'interaction comme critères pour le mécanisme de fusion. Notre support pour ce travail sera le système MATIS.

Références

[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.

[Coutaz 93]

Coutaz J., Faconti G., Paterno F., Nigay L., Salber D. MATIS: a UAN description and Lesson Learned. ESPRIT BRA 7040 Amodeus Project Document: System Modelling/WP14, Juin 1993.

[Hartson 92]

Hartson H.R., Gray P.D. Temporal Aspects of Tasks in the User Action Notation. Human Computer Interaction, Vol. 7, 1992, pp. 1-45.

[Lunati 91]

Lunati J.M. and Rudnicky A.I. Spoken Language Interfaces: The OM system. In Proc. CHI'91 Human Factors on Computing Systems, ACM Press, News Orleans, April 27-May 2 1991, pp. 453-454.

[Nigay 91]

Nigay L. et Coutaz J. Building User Interfaces: Organizing Software Agents. Proc. ESPRIT'91 Conference, Bruxelles, Nov. 1991, pp. 707-719.

[Nigay 92]

Nigay L. et Coutaz J. Interfaces multimodales et architecture : Fusion et parallélisme. Dans les actes de IHM'92 Quatrième journées sur l'ingénierie des interfaces homme-machine, Paris, 30 Nov., 1 et 2 Déc. 1992, pp. 29-36.

[Nigay 93]

Nigay L. and Coutaz J. A Design Space For Multimodal Systems: Concurrent processing and Data Fusion. In Proc. of INTERCHI'93, Amsterdam, Avril, 1993, pp. 172-178.

[Norman 86]

Norman D.A., Draper S. W. User Centered System Design, New Perspectives on Human-Computer Interaction. Lawrence Erlbaum Associates Publishers, 1986, 526 pages.

[UIMS 92]

The UIMS Tool Developers Workshop : A Metamodel for the Runtime Architecture of an Interactive System. In SIGCHI bulletin, ACM Press, 24 (1), January 1992, pp.32-37.