

Un environnement générique pour la réalisation d'interfaces multimodales

Un exemple d'application

Zhili ZHOU, Franck TARPIN-BERNARD, Bertrand DAVID

Laboratoire GRACIMP
ECOLE CENTRALE DE LYON
Département MIS
36, avenue Guy de Collongue
B.P. 163 69131 Ecully Cedex
Tél: 72.18.64.43 Fax: 78.33.16.15
E-mail: zhou@cc.ec-lyon.fr, tarpin@cc.ec-lyon.fr, david@cc.ec-lyon.fr

Résumé

Devant les besoins grandissant en interactions homme-machine performantes, on cherche de plus en plus à concevoir des applications proposant une interface multimodale. En effet, pour les applications à interface utilisateur complexe, l'interface multimodale propose différentes modalités permettant une meilleure prise en compte des différents aspects du dialogue. Cet article présente un environnement générique pour la réalisation d'interfaces multimodales. Cet environnement repose sur un éditeur de bibliothèques contextuelles d'événements monomodaux et multimodaux et sur un moteur de fusion des événements. Ce moteur peut être incorporé dans les applications sans remettre en cause leurs architectures. Un éditeur de schéma cinématique (CinémaTek) est aussi présenté en tant qu'exemple d'application produite dans cet environnement.

Mots clés

interface multimodale, système générique, événement monomodal et multimodal, bibliothèque contextuelle d'événements.

Abstract

With the growing needs for effective human-computer interactions and the appearance of new medias, we increasingly search to conceive applications with a multimodal interface. Indeed, for applications with complex interactions, multimodal interfaces propose different modalities that take in account the multiple characteristics of dialogue. This paper presents a generic environment for the realization of multimodal interfaces including an editor of contextual events library and a fusion engine. This engine can be incorporated in applications without changing the software architecture. An editor of cinematic schema (CinemaTek) is also presented as an application of this environment.

Key words

multimodal interface, generic system, monomodal and multimodal event, contextual events library.

I. Introduction

Depuis quelques années grâce à l'apparition de nouveaux médias dans la communication Homme-Machine (gants numérique, oculomètres, systèmes de reconnaissance et de synthèse vocale, etc.), il est intéressant d'imaginer de nouveaux modes de communication qui permettent aux utilisateurs d'interagir plus naturellement et plus rapidement avec l'ordinateur. L'interaction multimodale se fonde sur l'utilisation éventuellement simultanée des divers canaux de communication. La plupart des interfaces homme-machine multimodales actuelles font appel à l'entrée vocale et au geste de désignation. De nombreuses méthodes ont déjà été étudiées et certaines de leurs applications ont notamment été présentées dans les journées sur l'ingénierie des IHM (IHM'93, IHM'94 et IHM'95) [IHM 94]. Une synthèse est proposée dans [IHM 94].

L'objectif de notre travail est de fournir un système générique pour la réalisation d'interfaces multimodales. Celui-ci peut être intégré dans les applications d'une manière modulaire et contextuelle.

L'architecture générale de notre système et en particulier la composante reconnaissance de gestes sont présentées. Ensuite, nous précisons la définition des événements monomodaux et multimodaux sur laquelle repose le principe de fusion des événements. Finalement, un éditeur de schéma cinématique multimodal (CinémaTek) est présenté comme une application de notre système.

II. Architecture générale d'une application multimodale

Dans cet article nous utilisons la terminologie proposée par [BELLIK 92] à savoir : les périphériques d'acquisition fournissent au système des informations brutes (ex: coordonnées de souris, signal vocal échantillonné, etc.), une fois reconnues, ces informations deviennent des événements monomodaux, pour être ensuite fusionnées en événements multimodaux significatifs (on parle alors d'énoncés).

Toute application multimodale construite avec notre système est basée sur l'architecture qui comporte donc trois composants principaux: Le "Traitement des Informations d'Entrée", La "Fusion des Evénements Multimodaux" et "L'Interprétation d'un Evénement Multimodal" (cf. figure 1).

Lorsque l'application a reçu les informations en provenance des dispositifs d'entrée, le composant "Traitement des Informations d'Entrée" effectue tout d'abord des traitements préliminaires (filtrage, lissage, etc.). Ensuite il essaie de reconnaître les événements monomodaux correspondants en s'appuyant sur des événements monomodaux prédéfinis se trouvant dans la bibliothèque.

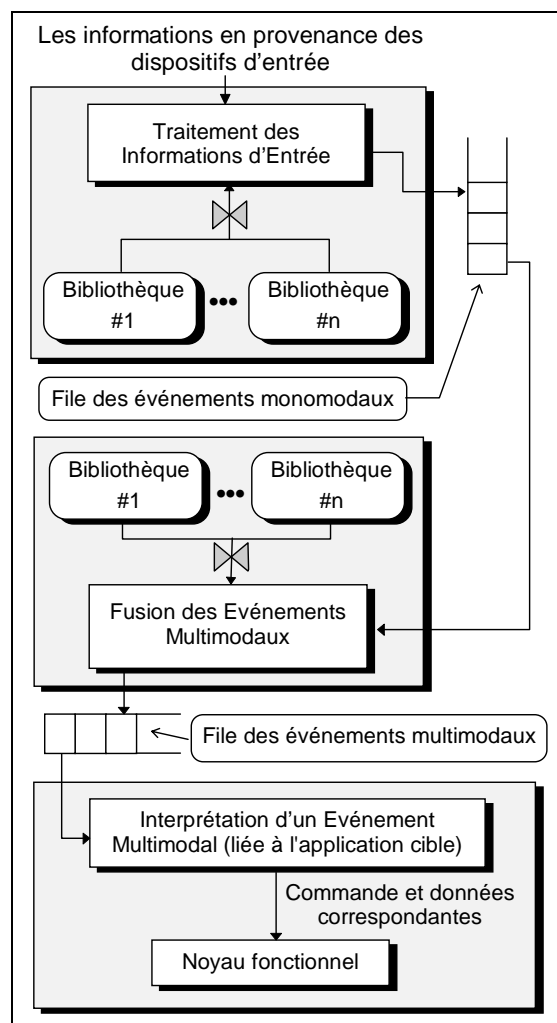



Figure 1: L'architecture générale d'une application multimodale

Le rôle du composant "Fusion des Événements Multimodaux" est d'exploiter le contenu de la file des événements monomodaux en consultant les prototypes d'événements multimodaux prédéfinis afin d'obtenir un événement multimodal effectif.

Une fois un événement multimodal identifié, celui-ci est envoyé au module "Interprétation d'un Événement Multimodal". Ce module active la commande correspondante en lui associant les paramètres nécessaires.

Dans les deux premières phases, pour obtenir un événement valide, nous devons comparer les informations reçues avec les prototypes d'événements (monomodaux ou multimodaux) prédéfinis. Dans les applications réelles, il se peut que la bibliothèque contienne des centaines, voire des milliers d'événements prédéfinis. Si l'on scrute tous les événements dans la bibliothèque, ceci va conduire à une augmentation sensible du temps de réponse. D'autre part, il est possible que le même événement soit défini porteur de significations différentes selon de contexte d'utilisation, dans le souci de limiter les efforts de mémorisation et d'apprentissage de l'utilisateur.

Ceci nous a donc amené à envisager de limiter le nombre d'événements possibles à un instant donné et de gérer la notion de "contexte" d'utilisation. Dans ce but, nous ajoutons une sélection contextuelle () permettant de limiter à bon escient le nombre de prototypes devant être examinés. *Une bibliothèque contextuelle* d'événements monomodaux ou multimodaux est un ensemble de prototypes d'événements prédéfinis groupés conformément au contexte des opérations.

III. Traitement des Informations d'Entrée

Dans sa version de base, nous prenons en compte trois types d'informations d'entrée: le click-souris, la parole et le geste.

Pour la désignation ou la sélection, il s'avère que le click-souris est une modalité appropriée [DAVID 93]. Lorsque un click a été produit, les coordonnées du point sont stockées tels quels dans la file d'événements monomodaux.

La reconnaissance de la parole est assurée par un système commercialisé de reconnaissance de mots isolés, tandis que la reconnaissance du geste est assuré par un système original développé au laboratoire [ZHOU 95]. Les algorithmes de reconnaissances de geste peuvent être classés en trois grands groupes: Les méthodes générales qui sont applicables dans un grand nombre de situations (de l'étude d'échantillons statistiques à la reconnaissance de formes), les méthodes connexionnistes qui utilisent les techniques des réseaux neuronaux et les méthodes structurelles qui se basent sur la structure des informations pour permettre la reconnaissance [SPIRITO 93]. La méthode qu'utilise ce dernier fait partie du troisième groupe d'algorithmes et est destinée à la reconnaissance de gestes effectués d'un seul trait à orientation imposée ou non sans contrainte de dimension. Du point de vue de sa structure un geste est une succession de segments et de courbes. La reconnaissance s'effectue sans apprentissage préalable.

Ce système de reconnaissance du geste est basé sur l'utilisation successive de plusieurs moteurs d'analyse (cf. figure 2).

Lorsque le système a reçu les points bruts issus de la saisie du geste, il déclenche le moteur "Prétraitement" qui améliore l'information obtenue (lissage, filtrage, des points d'appui). Puis le moteur "Optimisation des segments" identifie des segments élémentaires. A partir de ces segments, le moteur "Détection de la courbe" construit les morceaux de courbe en consultant les prototypes contenus dans la bibliothèque contextuelle. Le rôle du moteur "Vérification du geste" est de finir la comparaison entre la structure construite et les prototypes. Ce moteur évalue la correspondance entre les événements obtenus et ceux du prototype de la bibliothèque en éliminant, si c'est nécessaire, les petits éléments parasites pouvant être encore présents. La

reconnaissance structurelle ne s'effectue donc pas a priori, mais en fonction des gestes susceptibles d'être produits. Chaque moteur dispose d'une base de règles, et chaque règle possède un seuil pouvant influencer sur le taux de reconnaissance.

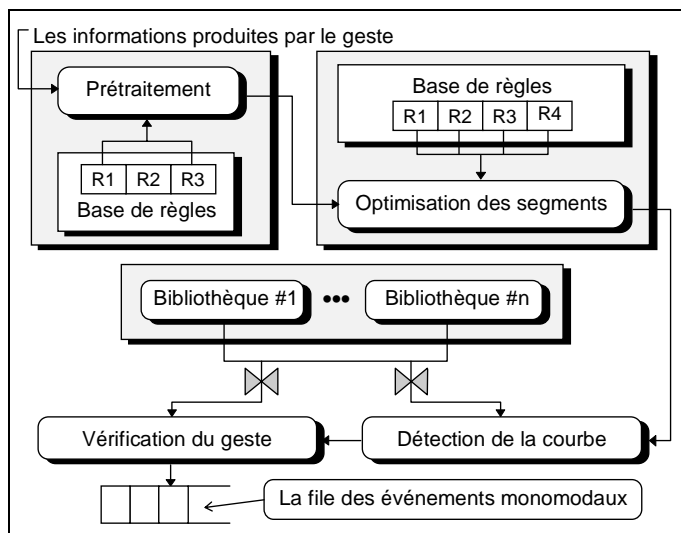


Figure 2: La structure du système de reconnaissance du geste

IV. Définition des prototypes d'événements

Du point de vue de sa structure, une interaction est basée sur trois composants (éventuellement facultatifs) qui sont: Action, Objet, Paramètre [DAVID 93]:

Action : correspond à une opération généralement appliquée à un objet. Le nombre d'actions dans un contexte donné est fixé à la conception de l'application.

Objet : exprime la portée de l'action. Il peut être explicite ou implicite, et unique ou constitué par un groupement. Le nombre d'objets est quelconque.

Paramètre : apporte un complément d'information à l'action ou à l'objet.

Ces trois notions ont été choisies pour caractériser les événements monomodaux.

Notre système fournit un éditeur pour générer la bibliothèque contextuelle de prototypes d'événements. La définition générale que nous adoptons pour un événement monomodal est la suivante:

- Nom de l'événement: identificateur textuel
- Type de la modalité: Geste, Click-souris ou Parole
- Natures de l'événement: Action, Objet, Paramètre ou combinaison
 - Identificateur textuel et numérique d'Action: (s'il existe)
 - Identificateur textuel et numérique d'Objet: (s'il existe)
 - Identificateur textuel et numérique de Paramètre: (s'il existe)
- Contenu physique: (variable suivant la modalité)
- Contexte élémentaire de fusion: oui/non
 - Si oui : ensemble d'associations :
 - . Nature: Objet ou Paramètre
 - . Mode: Nécessaire ou Par défaut
 - . Nombre d'événements de cette nature

Le champ "*Natures de l'événement*" indique la nature de l'événement monomodal et son identificateur textuel et numérique. Exemple: Pour un geste permettant de dessiner un rectangle aux coins arrondis (geste "round"), on a nature de l'événement *Action*: "dessiner", et également nature *Objet*: "l'objet dessinée" (le "round"). Il peut donc être défini ainsi: (cf figure 3)

Le champ "*Contexte élémentaire de fusion*" indique si un événement de ce type nécessite d'autres événements pour avoir un minimum de sens. Si c'est le cas, il faut indiquer les éléments nécessaires ou par défaut ainsi que leur nombre. Le "*Mode*" "nécessaire" indique que cet événement ne peut pas être effectif sans être fusionné avec un autre

événement de la nature correspondante. Le "Mode" "par défaut" indique que l'on peut se passer de cette valeur et prendre la valeur prédéfinie. Dans l'exemple précédent, pour un geste permettant de dessiner un objet (comme le geste "round"), on peut associer un paramètre correspondant à la couleur. Naturellement, ce paramètre peut prendre une valeur par défaut (couleur prédéfinie). Dans le cas d'un geste de suppression, un autre événement correspondant à l'objet à effectuer est nécessaire.

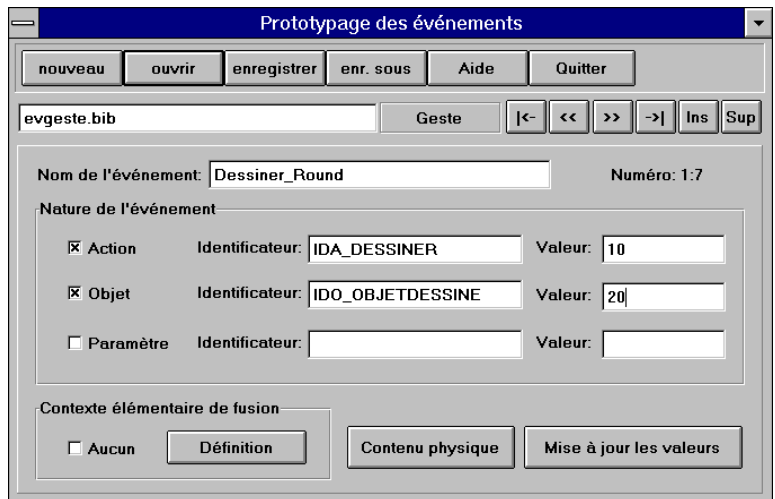


Figure 3: Définition d'un prototype d'événement monomodal

Le "Contenu physique" contient la description physique d'un événement. Pour un événement de type geste, c'est un ensemble de gestes élémentaires. Par exemple, le contenu physique du geste "round" comprend un segment vertical, une courbe et un autre segment horizontal (cf. figure 5). Pour un événement de type parole, le contenu physique contient les paramètres significatifs pour le système de reconnaissance. Dans notre cas, comme le système de reconnaissance est un système externe, le contenu physique est simplement la chaîne de caractère identifiant le mot isolé.

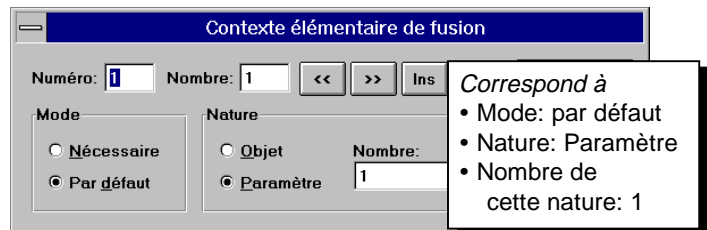


Figure 4: Définition du contexte élémentaire de fusion à un événement monomodal

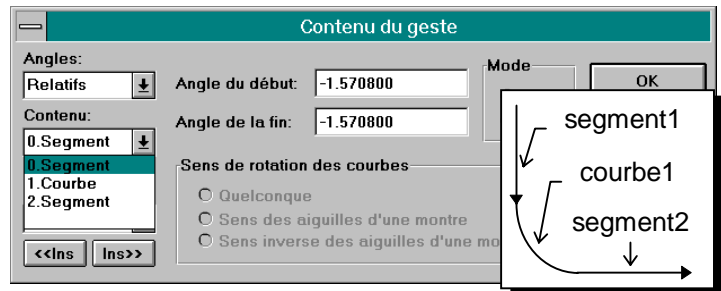


Figure 5: L'écran de la définition du contenu du geste ("round")

V. Fusion des événements monomodaux

Pour construire un événement multimodal effectif, l'ensemble des événements monomodaux qui le définissent doit contenir au moins l'Action et l'Objet sur lequel porte cette action, i.e. un événement multimodal effectif exige une Action, un ou plusieurs Objets, et peut avoir un ou plusieurs paramètres. Donc, la première tâche à effectuer est de trouver un événement "Action" dans la file des événements monomodaux. Ensuite, le système analyse le champ "Contexte élémentaire de fusion" de cet événement pour savoir de quels éléments celui-ci a besoin. Il recherche alors les événements correspondants dans la file monomodale (les événements choisis doivent être dans une fenêtre temporelle par rapport à la date de production de l'événement "Action"). Chaque événement ainsi choisi est alors confronté aux prototypes d'événements multimodaux de la bibliothèque contextuelle. Chaque élément de cette bibliothèque est en fait une association d'identificateurs d'action, objet et paramètre pouvant être fusionnés.

En pratique, la fusion des événements monomodaux repose sur la construction d'un arbre complet valide. L'événement "Action" est la racine de cet arbre. Le choix d'événements monomodaux satisfaisant le "Contexte élémentaire de fusion" de l'événement sélectionné permet de développer les branches de l'arbre (Cf. Figure 6). Nous avons choisi un ordre

de développement des noeuds en profondeur d'abord. Le processus de fusion s'arrête lorsque tous les contextes élémentaires de fusion des événements sélectionnés sont validés (tous les éléments nécessaires ont été trouvés et les éléments par défaut pouvant être retenus et se trouvant dans une certaine fenêtre temporelle ont été pris en compte).

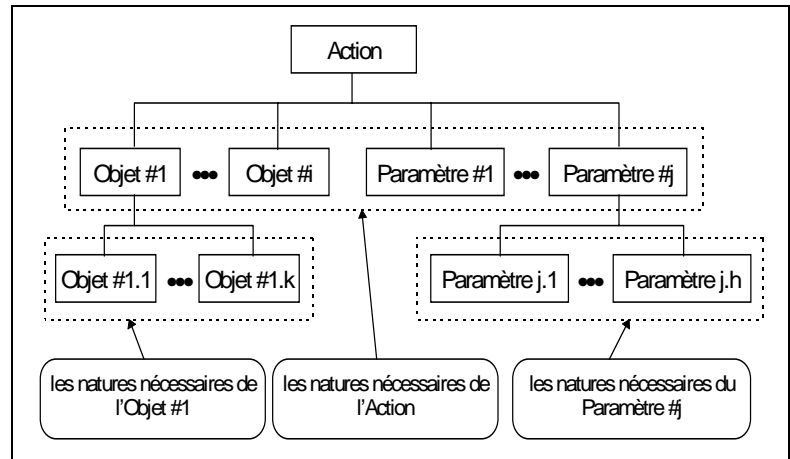


Figure 6: La fusion des événements monomodaux

Contrairement à une approche classique linéaire, cette stratégie n'impose pas de style de communication tel que Objet-Action-Paramètre ou Action-Objet-Paramètre. Les différents éléments peuvent provenir de modalités différentes dans n'importe quel ordre.

Illustration par un exemple du mécanisme de fusion :

Si un geste "round" a été produit, le système de fusion le retire de la file monomodale car c'est une Action. Il trouve en examinant son "Contexte élémentaire de fusion" que celui-ci peut être fusionné avec un événement "Paramètre". Il cherche alors dans la file des événements monomodaux ceux qui possèdent la nature "Paramètre". Il les confronte à la bibliothèque multimodale dans laquelle est indiqué que la

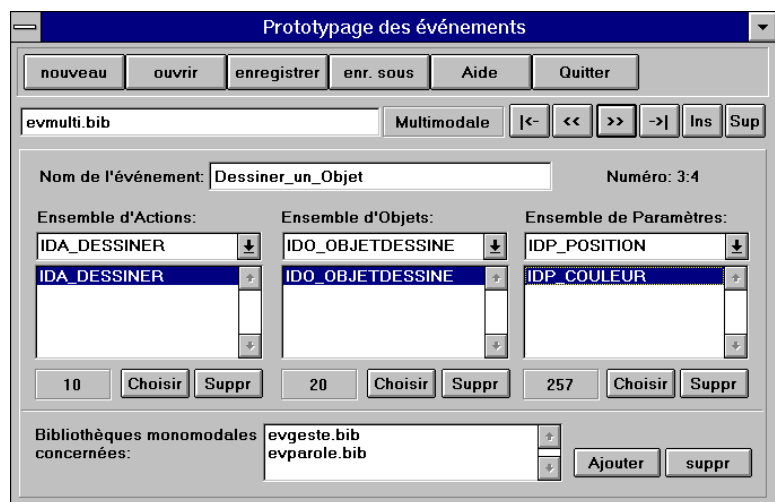


Figure 7: Définition d'un prototype d'événement multimodal

fusion entre le geste "round" et un événement est possible si ce dernier possède un paramètre ayant un identificateur IDP_COULEUR (cf. figure 7). Si le système en a trouvé un convenable dans la fenêtre temporelle correspondante, il le fusionne avec l'événement du geste "round", sinon celui-ci est stocké directement dans la file multimodale car le paramètre peut prendre une valeur par défaut.

Modification des coefficients et test de mise en oeuvre

Le système de reconnaissance du geste dispose de plusieurs moteurs d'analyse. Chaque moteur contient une base de règles et chaque règle a un seuil déterminant le résultat de reconnaissance. Dans les conditions réelles, les dispositifs utilisés et l'environnement de travail influent sur le taux de la reconnaissance. Il faut donc dans certains cas pouvoir ajuster ces seuils pour avoir un meilleur taux. Dans la version de base de notre environnement, cet ajustement se fait de façon manuelle (via une boîte de dialogue).

Lors de la construction d'un application, un test de mise en oeuvre est proposé pour tester les prototypes d'événements définis dans la bibliothèque.

VI. Un éditeur de schéma cinématique (Cinématek)

Dans le cadre de la génération de prototypes en mécanique, la CAO se doit de fournir aux concepteurs des outils qui leurs permettent de s'exprimer aussi librement que possible.

Cette remarque est particulièrement valable pour la construction de schémas cinématiques. Jusqu'à présent, les outils de création de schémas cinématiques reposent sur des interfaces homme-machine très éloignées de l'interface "naturelle" que constitue le papier et le crayon. Ainsi, le principe de saisie des schémas consiste généralement en la saisie textuelle des liaisons, de leurs positions et de leurs orientations ainsi que des noms des pièces qu'elles relie. Cette approche tient souvent au fait qu'il s'agit de produits spécialisés dans l'analyse et la validation de systèmes mécanique et non dans la construction de schémas en cours de conception.

Nous nous sommes donc intéressés à la réalisation d'un logiciel permettant une construction beaucoup plus naturelle et intuitive. Nous nous sommes donc efforcés de ne pas tomber dans le piège des éditeurs de CAO courants qui multiplient à outrance les menus rendant particulièrement difficile voire pénible la saisie d'informations.

Dans ce contexte, nous avons construit un éditeur appelé CinémaTek qui propose des fenêtres de visualisation (dans un repère 3D en projection parallèle) permettant la construction de schémas tri-dimensionnels (Cf. figure 8). L'édition d'un schéma cinématique s'effectue en construisant des squelettes de pièce, c'est-à-dire des ensembles de noeuds et de branches. Sur ces squelettes, on crée des liaisons par manipulation directe.

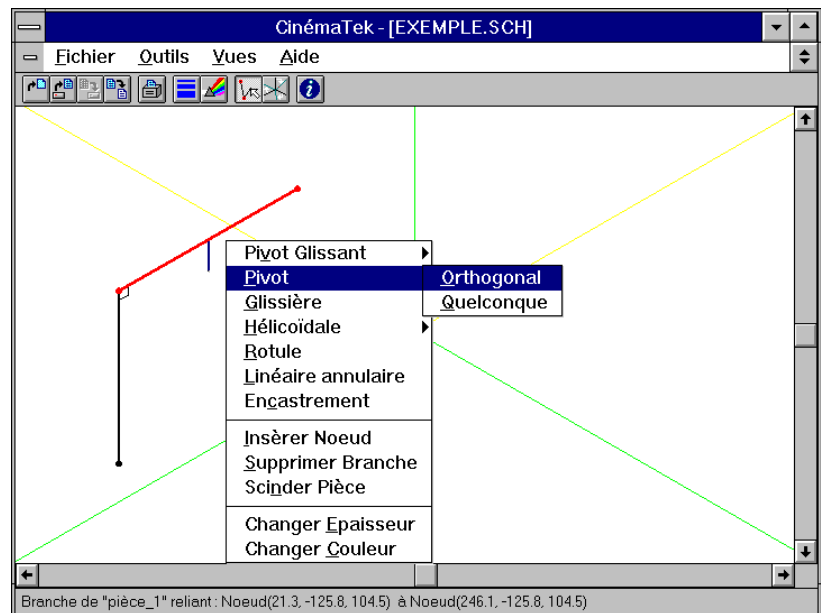


Figure 8: Construction d'une liaison sur une branche à l'aide du menu contextuel

Il est possible de renommer les pièces ou de les mettre en évidence en utilisant des couleurs ou des épaisseurs de traits différentes. Toutes les fonctions d'édition (création, modification, déplacement et suppression) sont possibles aussi bien sur les noeuds et les branches que sur les pièces ou les liaisons. Toutes les phases de création, ajout ou déplacement sont dynamiques, ce qui signifie que le schéma se transforme en temps réel au fil des actions de l'utilisateur sur le pointeur d'écran. Dans nos essais, ce pointeur a été manipulé soit avec la souris, soit avec le crayon d'une tablette graphique.

L'objectif principal de notre éditeur est de fournir un système souple permettant au concepteur de traduire très rapidement ses idées sous forme de schémas directement utilisables. Pour faciliter le maniement du pointeur dans l'espace 3D, les mouvements de ce dernier sont analysés par rapport aux trois directions principales que constituent les axes du repère. Ces directions sont bien sûr fonctions de la position de l'observateur du schéma. Cette méthode permet de proposer un système dans lequel on ne retrouve pas les 3 fenêtres classiques de projection sur les plans principaux du repère. Ceci permet une saisie beaucoup plus rapide dans un contexte où la précision de positionnement des objets n'est pas fondamentale. Cependant, pour assister le concepteur dans sa perception de la profondeur de la scène, nous visualisons des cubes de projection pour chaque noeud de la pièce en cours de construction. Par ailleurs, il est possible de créer simultanément d'autres fenêtres de visualisation pour observer le schéma sous d'autres angles. Il est aussi possible de visualiser plusieurs schémas en même temps. Enfin, le système détecte automatiquement les perpendicularités entre branches qu'il traduit au fur

et à mesure sous forme de petits parallélogrammes. De fait, CinémaTek permet une expression simple d'intentions conceptuelles telles que la perpendicularité ou le parallélisme. Sous cette forme, CinémaTek permet donc une saisie rapide et beaucoup plus naturelle. La souplesse de ses fonctionnalités de modification permet au concepteur de tester immédiatement des solutions différentes. Dans une première version, la création puis la modification des éléments s'effectuait via des menus contextuels apparaissant à l'emplacement de la souris (cf. figure 8).

Après avoir construit une interface basée sur la métaphore papier-crayon, nous avons cherché à pousser la métaphore de manipulation directe encore plus loin en ajoutant à CinémaTek le système de reconnaissance et de fusion d'événements multimodaux décrit précédemment. Grâce à ce système, et en utilisant le crayon de la tablette graphique pour plus de facilité, la construction des liaisons du schéma cinématique ne se fait plus seulement par sélection dans des menus contextuels, mais peut aussi être réalisée à l'aide de gestes basés sur les représentations 2D standardisées des liaisons (cf. figure 9).

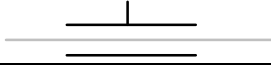

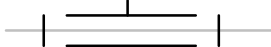

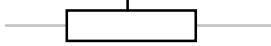
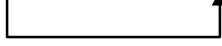
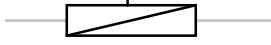


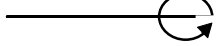




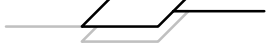

Nom de la liaison	Représentation 2D standard	Geste dans CinémaTek
Pivot Glissant		
Pivot		
Glissière		
Hélicoïdale		
Linéaire Annulaire		
Rotule		
Ponctuelle		
Appui Plan		

Figure 9: L'ensemble des prototypes d'événement gestuel

Ces gestes sont regroupés dans des bibliothèques activées en fonction des différents contextes de l'application (élément sélectionné: noeud, branche, liaison, pièce). Ainsi les gestes de définition de liaisons ponctuelles ou appuis plan ne peuvent être réalisés qu'après la sélection d'un des noeuds du squelette.

De même, nous étudions les avantages que peut apporter le système de reconnaissance vocale, notamment dans l'expression des données techniques du domaine. Pour l'instant, nous utilisons le canal vocal pour indiquer le paramètre d'orthogonalité des liaisons. La figure 10 présente un squelette de pièce en cours de construction après qu'une liaison pivot orthogonal ait été construite. La commande d'insertion a été déclenchée en effectuant le geste "Liaison Pivot" sur l'arbre concerné et en prononçant le mot clé "orthogonal". Le paramètre d'orthogonalité prend la valeur par défaut "quelconque", s'il n'est pas produit par une des modalités utilisables.

De fait, le concepteur peut se concentrer sur sa tâche de conception en utilisant un crayon et en "dessinant" ses schémas sans avoir à se préoccuper des contraintes de l'éditeur. Les intentions conceptuelles élémentaires (orthogonalités, directions principales, etc.) sont automatiquement prises en compte par CinémaTek car le mode d'expression utilisé pour la construction du schéma transpose implicitement ces intentions.

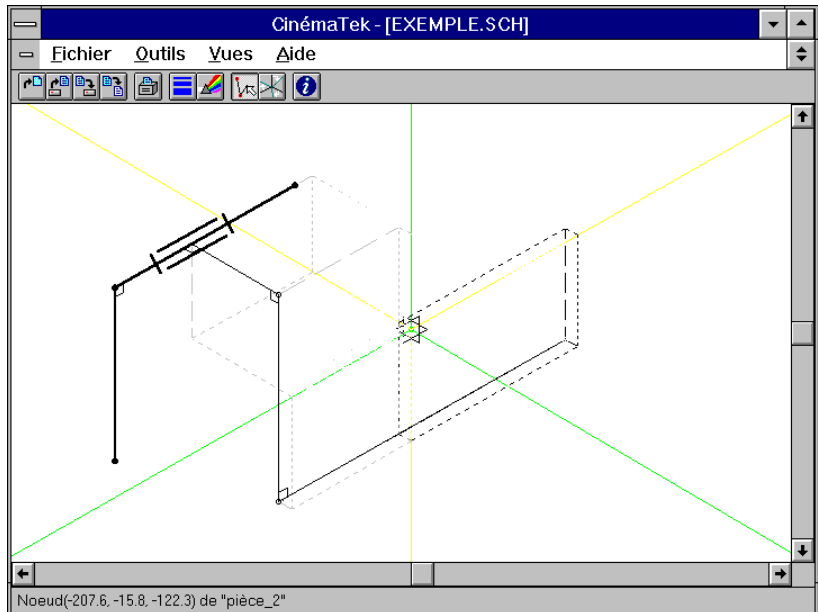


Figure 10: Exemple de construction

VII.Conclusion

Nous avons présenté un environnement générique pour la réalisation d'interfaces multimodales ainsi qu'une application construite avec celui-ci. Cet environnement contient un ensemble d'outils spécialisés permettant la création des bibliothèques associées à chaque modalité et la définition des constructions multimodales valides. Par exemple, pour le geste, il s'agit d'acquérir, pour chaque contexte identifié, l'ensemble des gestes valides, de tester leur niveau de reconnaissance (notamment en terme de séparabilité), d'ajuster les seuils de reconnaissance, et de générer les bibliothèques d'événements monomodaux et multimodaux. Tous les seuils des règles utilisées par les systèmes de reconnaissance sont modifiables pour s'adapter aux applications réelles. Dans certains cas, le concepteur d'applications peut définir ses propres règles et les ajouter dans les bases de règles.

La reconnaissance des événements monomodaux et leur fusion repose principalement sur des constructions progressives contextuelles. La considération temporelle tient une grande place dans notre méthode de fusion des événements monomodaux.

Actuellement, cet environnement a été appliqué à un éditeur de schéma cinématique (CinémaTek), Le résultat est satisfaisant surtout au niveau de la reconnaissance du geste.

VIII.Bibliographie

- [BELLIK 92] *Yacine Bellik, D. Teil, Définitions terminologiques pour la communication multimodale*, IHM'92, Décembre 1992
- [CHATTY 95] *Stéphane Chatty, Patrick Lecoanet, Un poste de travail avec reconnaissance de gestes pour le contrôle aérien* IHM' 95, P81-P88, Octobre 1995
- [DAVID 93] *Bertrand DAVID, Salah SADOU, Patricio SPIRITO, Mourad DJEBALI, Sélection des Modalités pour une Interface Multimodale*, IHM' 93, P37-P44, Octobre 1993
- [IHM 94] **Systèmes d'Analyse des Interactions Homme-Ordinateur**, Synthèse de travaux de l'atelier "Interfaces multimodales" d'IHM'93, Actes d'IHM'94, Lille, 1994, p.243-298
- [RUBINE 92] *D. H. Rubine, Combining Gestures & direct manipulation*, CHI'92, May 3-7, 1992
- [SPIRITO 93] *SPIRITO Patrizio, Interfaces Gestuelles à Reconnaissance Précoce*, Rapport de DEA, Ecole Centrale de Lyon, Septembre 1993
- [ZHOU 95] *Zhili ZHOU, Etude de l'Interface Homme-Machine Multimodale*, Rapport de DEA, Septembre 1995