



HAL
open science

Conception assistée par ordinateur d'aménagement d'une cuisine

Madeleine Arnold, Sandrine Robbe, Bellik Yacine, Eric Brison, Jean Caelen,
Claudie Faure, Laurent Guittet, Denis Lalanne, Nigay Laurence, Nadine
Vigouroux

► To cite this version:

Madeleine Arnold, Sandrine Robbe, Bellik Yacine, Eric Brison, Jean Caelen, et al.. Conception assistée par ordinateur d'aménagement d'une cuisine. Journées Francophones sur l'Ingénierie de l'Interaction Homme-Machine (IHM 1996), Association Francophone d'Interaction Homme-Machine (AFIHM), 1996, Grenoble, France. pp.105-121. hal-03608367

HAL Id: hal-03608367

<https://hal.science/hal-03608367>

Submitted on 30 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Conception assistée par ordinateur d'aménagement d'une cuisine (Rapport de l'atelier)

Rédacteurs

*Madeleine Arnold, Sandrine Robbe,
Yacine Bellik, Eric Brison,
Jean Caelen, Claudie Faure,
Laurent Guittet, Denis Lalanne,
Laurence Nigay, Nadine Vigouroux*

emails:

*arnold@idf.ext.ju, balbo@loria.fr,
bellik@limsi.fr, brison@irit.fr,
caelen@imag.fr, cfaure@sig.enst.fr,
guittet@ensma.univ-poitiers.fr,
lalanne@imt.dmt.epfl.ch,
nigay@imag.fr, vigourou@irit.fr*

Participants

*Cédric Dumas, Ludovic Imberdis
Xavier Poinard, Franck Poirier, Daniel Teil*

emails:

*dumas@lifl.fr, imberdis@imag.fr,
xavier.soinard@der.edfgdf.dr,
poirier@iu-vannes.fr,
teil@limsi.fr*

RÉSUMÉ

Cet article rapporte les travaux de l'atelier sur l'étude de cas : conception assistée par ordinateur d'aménagement d'une cuisine. Le travail réalisé a consisté à concevoir le système interactif. L'étude de cas est un cadre de réflexion commun dans lequel les méthodes, modèles et outils de conception des participants sont confrontés. Dans ce rapport, nous présentons tout d'abord les étapes du processus de conception adopté. Nous discutons ensuite la prise en compte des contraintes, les modèles de tâche pour les logiciels de CAO. Enfin des techniques d'interaction et des approches sur la résolution de référence et de déictique sont présentées.

MOTS CLÉS : processus de conception, CAO, contraintes, modèle de tâche, activité de l'utilisateur, interaction au stylo, multimodalité et déictique.

INTRODUCTION ET OBJECTIF

Les activités de l'atelier concernent une étude de cas : la conception assistée par ordinateur d'aménagement d'une cuisine. Le sujet de l'étude est un logiciel qui permet de concevoir l'aménagement d'une cuisine à partir d'un ensemble de meubles défini et produit par les constructeurs de cuisine.

L'objectif de cette étude de cas est de définir un cadre de réflexion commun dans lequel les méthodes, modèles et outils de conception utilisés par les participants sont confrontés. Aussi les travaux de l'atelier ont concerné toutes les étapes du processus de conception d'un système interactif pour une tâche d'aménagement de cuisines.

Avant de présenter les résultats de l'atelier, il est nécessaire de fixer certains paramètres de l'étude de cas. Nous avons tout d'abord décidé que le système à concevoir n'exploite pas nécessairement un système

existant comme un système de CAO (Conception Assistée par Ordinateur). Par ce choix, nous ne limitons pas la conception et considérons que tout est à concevoir et à développer. Ce choix ne nous a néanmoins pas empêché d'étudier et de s'inspirer de l'existant comme dans [1] et [17]. L'autre paramètre de l'étude à fixer sont les utilisateurs du système. Pour cela nous avons identifié quatre situations d'interaction présentées à la figure 1.

Comme le montre la figure 1, le premier système envisageable (cas 1) est destiné au cuisiniste. Il est le seul à utiliser le système. Les cuisinistes utilisent des systèmes informatiques pour l'établissement d'un devis à partir d'une liste de meubles et d'électroménagers sélectionnés. Néanmoins la conception de l'aménagement doit être faite avant d'utiliser de tel système.

Dans le deuxième cas de la figure 1, nous envisageons un système où le cuisiniste interagit avec le système en présence des clients qui peuvent lui demander de faire-faire une action.

Le troisième cas est sans doute le plus général : il s'agit d'un système où le cuisiniste et le client travaillent ensemble à l'aménagement de la cuisine.

Enfin le dernier cas envisagé correspond à un système qui remplacerait les compétences du cuisiniste et permettrait au client de concevoir sa cuisine seul. Nous pouvons même envisager un tel système accessible via le réseau : le client sans aller au magasin conçoit le plan de sa cuisine.

Dans la suite, nous avons décidé de considérer le cas 3 où le cuisiniste (CU) et le client (CL) travaillent ensemble à l'aménagement de la cuisine. Les modalités de communication utilisées par le cuisiniste concernent le faire-savoir et le faire-faire. En vue de la différence d'expertise entre le cuisiniste et le client, certaines fonctions du système ne seront dédiées qu'au cuisiniste, d'autres qu'au client. De plus l'interface du système ne

doit pas être conçue de la même manière : des techniques d'interaction différentes doivent être envisagées pour le cuisiniste et pour le client.

Cet article présente les solutions et problèmes identifiés au cours de l'atelier. Plusieurs étapes de conception sont considérées. Dans une première partie, nous exposons les étapes de la méthode de conception adoptée. Ce processus de conception structure la suite du rapport. Nous rapporterons donc ensuite les résultats et problèmes de conception pour chaque étape, de l'étude de l'activité de l'utilisateur à la conception logicielle du système interactif.

**APPROCHE DE TRAVAIL :
MÉTHODE DE CONCEPTION**

Pour organiser le travail en atelier, nous avons tout d'abord adopté une méthode de conception qui permet de structurer en plusieurs phases la conception du système

interactif. Deux soumissions [17] [21] présentent des méthodes de conception dans le cadre de l'étude de cas.

Nous décomposons la tâche de conception en six sous-tâches essentielles : esquisse du problème à résoudre, identification de l'utilisateur, définition des tâches et des objets spécifiques au domaine d'application, définition des objets et fonctions informatiques, définition de la présentation de l'interface, et évaluation expérimentale. Ce schéma est une vue simplifiée du processus de conception. En pratique, les sous-tâches sont chacune soumises à des révisions accompagnées de nombreux retours arrière. Dans l'atelier nous avons étudié les différentes phases, certaines ayant été plus approfondies que d'autres. Nous n'avons pas abordé la conception logicielle ni la phase de codage. Les sous-paragraphes suivant décrivent le rôle de chaque phase de conception.

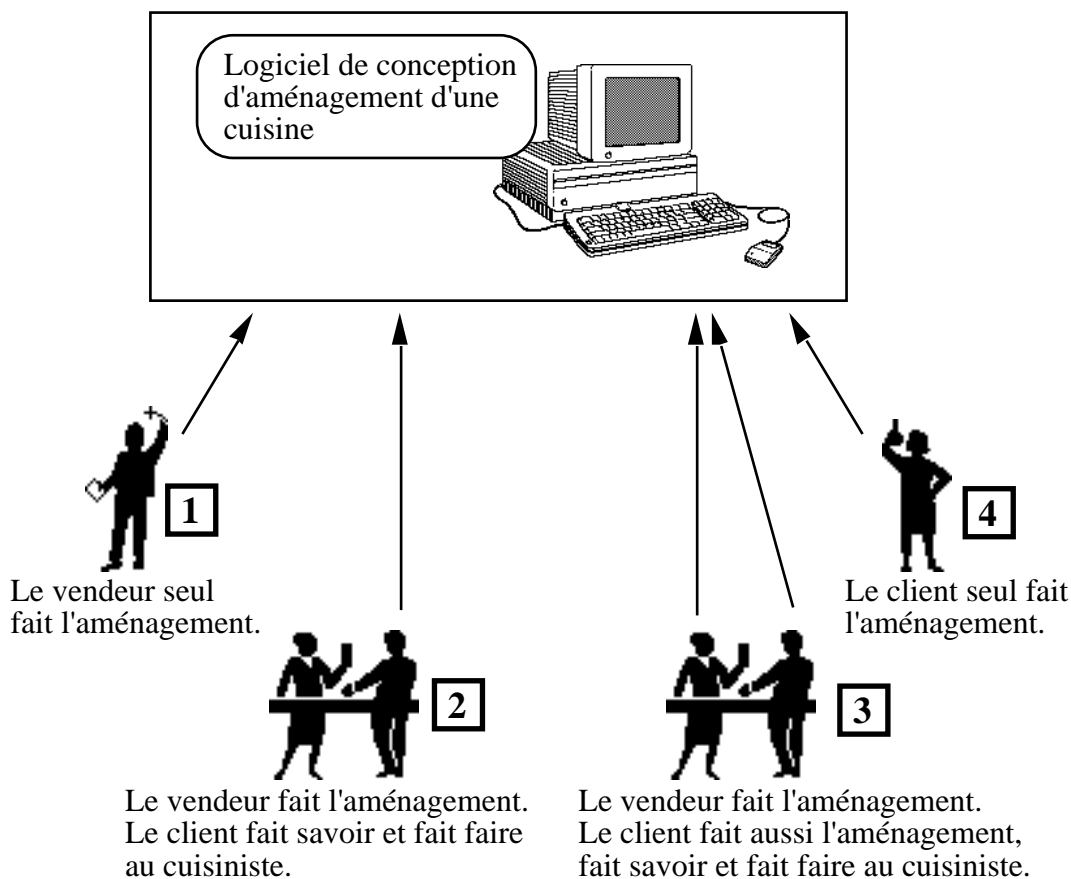


Figure 1 : Quatre situations d'interaction.

Identification de l'utilisateur

La définition d'un modèle de l'utilisateur type s'appuie généralement sur des données informelles recueillies au moyen de questionnaires, de discussions, d'enregistrements vidéo [1]. L'analyse de ces données est un art difficile que les informaticiens ne maîtrisent pas toujours. Un point de départ possible est une classification générale en utilisateurs novices, experts

et occasionnels. A cette dimension, il faut ajouter la catégorie socioprofessionnelle et le type de connaissances (connaissances dans le domaine d'application, connaissances dans le domaine informatique).

Comme nous l'avons décrit, deux types d'utilisateur sont identifiés : le client (CL) et le cuisiniste (CU). Le paragraphe 3 décrit notre analyse de l'utilisateur.

Définition des tâches et des objets du domaine applicatif

La définition des tâches et des objets du domaine d'application s'effectue sans considération aucune pour la dimension informatique et pour les techniques d'interaction.

Les objets correspondent aux variables psychologiques des utilisateurs. Ils décrivent les notions propres au domaine. Des opérateurs, qui constituent les tâches élémentaires du domaine, leur sont associés. Une fois les objets et leurs opérateurs introduits, il faut identifier les tâches et déterminer leurs relations. La technique usuelle consiste à décomposer une tâche en sous-tâches jusqu'à la rencontre des tâches élémentaires : une approche descendante par affinements successifs.

Cette analyse repose sur l'observation d'une situation réelle de travail afin d'identifier les variables psychologiques des utilisateurs : CL, ayant des idées plus ou moins précises de son besoin, met en forme un projet d'aménagement de cuisine, avec l'aide d'un spécialiste du métier, CU. L'étude de l'activité permet de dégager trois objectifs prioritaires du système :

- une conception itérative du produit permettant l'expression de choix dans divers domaines (géométrie, esthétisme, fonctionnalité, prix, disponibilité, etc.), conservation de la trace, retours en arrière (choix exprimés par CL et/ou CU),
- une perception de la cuisine finale réaliste (surtout pour CL),
- la vente d'une cuisine à l'issue du premier rendez-vous (surtout pour CU).

Ces objectifs se traduisent en objets et tâches. De plus ces tâches seront dépendantes de l'utilisateur : celles effectuées par l'utilisateur et celles destinées au cuisiniste.

Définition des objets et fonctions informatiques

Le domaine cerné, il faut pratiquer sa projection en termes de concepts informatiques : objets et fonctions. Les objets se répartissent à leur tour en objets sémantiques et syntaxiques. Un objet sémantique modélise un objet du domaine. Un objet syntaxique participe à la réalisation de tâches syntaxiques. Une tâche syntaxique ne relève pas du domaine mais est induite par le support informatique. Par exemple, l'utilisation des systèmes de fenêtrage implique la manipulation de fenêtres, le défilement d'informations, ou permet d'ouvrir plusieurs vues sur un même document. Ces tâches sont dites syntaxiques car ne sont pas déduites du domaine applicatif : l'aménagement d'une cuisine.

Comme pour les tâches identifiées à la phase précédente, les fonctions du système seront dépendantes de l'expertise de l'utilisateur :

- le cuisiniste expert dans son domaine mais novice en informatique, pourra s'adapter rapidement au système : l'utilisation du système est quotidienne.
- le client dont l'"expertise" réside en la connaissance de son but, demande plus de "naturel" dans l'interaction avec le système. Ceci implique un accroissement significatif des fonctions du système comme par exemple la prise en compte de contraintes floues comme "une cuisine style rustique, assez claire, pour une famille d'au moins quatre personnes".

L'objectif de cette phase est double :

- **identification des objets du domaine et leurs relations;** Nous identifions deux objets principaux qui sont la pièce (modèle géométrique et technique) et l'ameublement. Ce dernier possède des attributs tels que le style et la couleur et se décompose en sous-objets : les meubles et appareils électroménagers. Tous ces objets sont liés par des relations géométriques et/ou topologiques (à côté de, sous la fenêtre, etc.).
- **identification des fonction;** celles-ci doivent permettre à l'utilisateur de réaliser les tâches identifiées à la phase précédente.

Citons le premier niveau de décomposition :

- +1 Elaboration du modèle géométrique de la pièce
- +2 Définition des éléments techniques
- +3 Prise en compte des caractéristiques de l'aménagement
 - a attributs
 - b positionnement
- +4 Exploitation du modèle
 - a visualisation
 - b métrages, nomenclature, devis, etc.
 - c vérification (ou validation) fonctionnelle (pas de hotte, ouvertures exclusives de portes,...).

L'ordre partiel identifié est le suivant : 1<2 1<3b 4bc > 3. Par contre, 4a peut être fait immédiatement après 3a et avant 1 et 2, comme dans le cas de l'utilisation d'aménagement prédéfini.

Définition de la présentation de l'interface

La présentation de l'interface définit le comportement perceptible du système. L'objectif est de choisir la présentation et la manipulation des objets sémantiques et syntaxiques identifiées à la phase précédente en sorte que l'utilisateur élabore un modèle conceptuel en accord avec le fonctionnement du système. Cette étape peut s'effectuer en s'appuyant sur deux règles essentielles : (1) tout objet sémantique et syntaxique doit avoir son

correspondant sous forme d'objet de présentation; et (2) l'état du système doit être explicite.

Les objets du modèle fonctionnel (mur, meuble, etc.) deviennent objets d'interaction grâce à leurs présentations. Les objets syntaxiques, qui ne sont pas propres au domaine, doivent aussi être présents comme un système de feuilletage du catalogue des cuisines. Pour guider la conception de la présentation et garantir une cohérence globale de l'interface, une métaphore d'interaction peut être choisie. Nous envisageons une métaphore du monde réel : crayon, règle et gomme apparaissent alors en tant qu'outils de dessin.

Lors de cette phase de conception, il est important de prendre en compte le lien entre les fonctions de présentation (en sortie) et celles d'acquisition (en entrée) de l'interface. Les objets de présentation sont des lieux d'interaction : des dispositifs physiques d'entrée et de sortie doivent être choisis. Par exemple la saisie des dimensions de la pièce peut se faire par manipulation directe avec la souris, à l'aide d'un scanner à partir d'un plan d'architecture mais aussi à l'aide d'un stylo [11, 12]. De même la première esquisse de la cuisine peut se faire avec un stylo à main levée.

La conception de l'interface peut se décomposer en deux sous-tâches :

- Une première étape dite "logique" consiste en un découpage fonctionnel correspondant à la hiérarchie des tâches identifiées. Ce découpage permet de structurer l'ensemble de l'interface.
- Une deuxième étape dite "physique" consiste à définir tous les objets de présentation : pour chaque objet, il convient de concevoir une présentation et une technique d'interaction adéquate.

L'expression des souhaits (ou définitions) du client exprimées sous forme de buts pose un problème intéressant d'interaction homme-machine. Au cours de cet atelier la nature et la prise en compte des contraintes ont été longuement discutées pour une activité de conception. La prise en compte des contraintes et les modèles de tâche ainsi que ceux de dialogue font l'objet des deux paragraphes suivants.

PRISE EN COMPTE DES CONTRAINTES ET DES CRITERES DANS UNE TÂCHE DE CONCEPTION

Le traitement des contraintes dans la conception est souvent lié à la satisfaction de contraintes, donc aux techniques de l'intelligence artificielle, et est bizarrement par la même occasion écarté du champ de l'IHM. Bien qu'elles soient à la base de l'interaction entre l'humain et la machine dans une tâche de conception, les contraintes, que nous diviserons en *contraintes* et *critères*, sont rarement des données directement manipulables par l'utilisateur.

Nous présentons la prise en compte des contraintes à tous les niveaux d'une tâche de conception, leur distinction avec les critères et leur importance dans la conception de l'interaction humain-machine. Nous souhaitons montrer que l'interaction humain-machine doit pouvoir avoir lieu au niveau du raisonnement de l'utilisateur. La conception est une tâche cognitive particulièrement complexe, qui requiert de la créativité et de nombreux calculs. Ce n'est donc ni une tâche facile pour le concepteur ni pour la machine. L'intelligence humaine en collaboration avec une intelligence artificielle devrait aboutir à une intelligence augmentée issue de leur synergie. C'est ce que nous souhaitons démontrer dans ce paragraphe en montrant que le traitement des contraintes appartient aussi bien au champ de l'intelligence artificielle qu'à celui de l'interaction homme-machine.

Définitions

Le distinguo entre contraintes et critères n'ayant pas été très clair lors de l'atelier, il semble nécessaire de le définir. Le petit Robert donne les définitions suivantes. Une contrainte est une règle obligatoire et pénible à appliquer. Alors qu'un critère est ce qui sert de base à un jugement d'appréciation. C'est un caractère, un signe qui permet de distinguer une chose, une notion, de porter sur un objet un jugement.

D'une façon plus sommaire, une contrainte est satisfaite ou non alors qu'un critère peut prendre toute une gamme de valeurs d'appréciation. Par exemple, si l'on considère le prix comme étant une contrainte, l'utilisateur pourra dire qu'il veut une cuisine ne dépassant pas 40.000 francs. A ce moment là, toutes les solutions dépassant 40.000 francs seront évitées. Par contre, si l'on considère le prix comme étant un critère, aucune solution ne sera interdite pour des raisons pécuniaires. Le concepteur aura la possibilité de comparer les solutions de cuisine en prenant en compte leur prix.

Intervention des contraintes et critères dans le processus de conception

Une contrainte floue est une règle peu précise. Néanmoins cela reste une contrainte alors qu'un critère permet d'évaluer. Ainsi, dans la conception, les contraintes permettront de définir, de contraindre, un espace de solutions et les critères permettront d'évaluer cet espace.

Pour que la notion de critère ait un intérêt il faut prendre en compte un raisonnement plus fin que celui des contraintes. Il n'y a pas de solution idéale à un problème, mais un ensemble de bonnes solutions. Certaines sont meilleures selon certains critères et d'autres le sont pour d'autres. C'est en raffinant incrémentalement sa définition des contraintes et des critères, en interagissant avec le système, que l'utilisateur va atteindre sa meilleure solution.

Par exemple, voici quelques attributs identifiés dans le cas de la conception d'une cuisine :

- topographiques
- environnementales (murs, plan, ...)

- financières (budget)
- esthétiques
- utilitaires
- temporelles (temps de construction, de livraison, ...)

Comme nous venons de l'expliquer, chaque attribut peut aussi bien être une contrainte qu'un critère ; c'est au choix de l'utilisateur. Cependant, il nous semble que les attributs liés à l'environnement sont le plus souvent des contraintes; il est difficile de modifier la forme de la pièce ou de ne pas tenir compte des meubles déjà existants. Il se peut néanmoins que des solutions créatives passent par la violation de ces contraintes. Supposons que le client veuille une cuisine spacieuse alors que sa pièce est petite, il faudra par exemple casser un mur pour satisfaire son critère. Il est important de laisser l'utilisateur libre de classer lui-même les attributs du noyau fonctionnel en contraintes et critères, sachant que l'interaction avec la machine l'amènera sûrement à réviser ses choix ou à en définir de nouveaux.

La notion de contraintes et de critères intervient à tous les niveaux de la conception. Elle peut être plus ou moins cachée à l'utilisateur selon l'expertise (CL ou CU), le contrôle que l'on souhaite lui allouer, la situation d'interaction (Voir Figure 1). Les contraintes sont généralement plus importantes dans les phases de définition de la conception alors que les critères interviennent plutôt dans les phases d'évaluation. Cependant, définition et évaluation doivent être dans une même boucle que l'utilisateur va parcourir de nombreuses fois. Ainsi définition et évaluation ne représenteront plus qu'un seul incrément dans un processus de conception.

Expression et visualisation des contraintes : un problème d'interaction Homme/Machine.

L'utilisateur doit pouvoir accéder à la définition des contraintes à chaque fois qu'il le désire. Ce qui signifie que la définition des contraintes n'est pas statique ; elle n'est pas présente juste une fois après l'analyse du concepteur. La définition est dynamique. Le concepteur va modifier son problème à chaque fois qu'il ne sera pas satisfait à des résultats, soit parce qu'il n'y a aucune solution (problème sur-contraint, nécessité de trouver les conflits), soit parce qu'il y a trop de solutions (problème sous-contraint, nécessité de choisir des critères pour les comparer ou de restreindre davantage la définition) soit parce qu'elles ne le satisfont pas. Néanmoins, c'est bien l'utilisateur qui fait intervenir la dynamique dans le réseau de contraintes. Il n'y a pas d'algorithme génétique, de réseau de neurones ou de système de raisonnement à base de cas permettant de faire muter le problème vers la bonne solution. Nous croyons au pouvoir de la visualisation et de l'interaction qui peut stimuler le concepteur et augmenter ses capacités dans la résolution de problèmes. Autrement dit, nous ne souhaitons pas rendre créatives les machines, nous pensons que c'est en supportant les lacunes de l'utilisateur, qu'elles

soient computationnelles, perceptuelles ou attentionnelles, que nous arriverons à créer une intelligence augmentée issue de la synergie homme-machine.

De la même façon, nous verrons par la suite que l'utilisateur doit pouvoir accéder à la définition des critères à chaque fois qu'il le désire. La gestion des critères est néanmoins différente de celle des contraintes ; elle nécessite l'existence de solutions, et est basée sur l'évaluation.

Notons, de plus, qu'il doit y avoir différents types d'interface entre l'utilisateur et le système de résolution de contraintes. Fournir à l'utilisateur une représentation sous forme logique des règles n'est pas suffisant car non intuitive. La visualisation est particulièrement importante dans la conception d'une cuisine. Il est donc important de soutenir l'utilisateur le plus tôt possible avec des visualisations plus ou moins réalistes du produit finale. Cependant, la représentation graphique n'est pas toujours adaptée pour définir certaines contraintes. Définir que le réfrigérateur doit être à moins de trois mètres de la table peut être difficile à définir graphiquement. Il est donc nécessaire de fournir à l'utilisateur un éditeur mixte, aussi bien graphique que textuel pour la définition de règles plus abstraites.

Nous ne discuterons pas ici des modalités qui peuvent être utilisées par l'utilisateur pour communiquer ses contraintes et ses critères à la machine. Remarquons, néanmoins, que certaines modalités peuvent s'avérer meilleures que d'autres en fonction de la précision de la définition. En ce sens, le chapitre suivant montre que chaque modalité peut jouer un rôle particulier dans une tâche de conception. Par exemple, il est possible d'imaginer une phase d'initialisation du système où l'utilisateur pourra esquisser à l'aide d'un crayon la cuisine qu'il désire. Cette phase graphique correspond à l'esquisse de la scène pour la mise en place des éléments sans qu'il soit nécessaire de les spécifier complètement. Il est aussi possible d'utiliser la voix, le geste ou la réalité virtuelle.

Architecture supportant l'analyse sous l'angle contraintes/critères

Ouverte et multiagents Pour libérer les machines de leur mode pré défini ou pour les enrichir de l'interaction avec la nature, nous devons permettre aux humains de les aider de la même façon que les humains profitent de la puissance de calcul des machines [23]. Nous proposons une approche qui intègre plusieurs systèmes intelligents dans une architecture ouverte et distribuée. Plus important encore, cette architecture facilitera la collaboration homme-machine et la coexistence, ainsi que la co-contribution, de la créativité humaine et artificielle. Une architecture ouverte est différente de la nature traditionnelle algorithmique des logiciels. Elle organise ses fonctionnalités au niveau de la distribution de tâche au lieu de la performance de tâche. Plus spécifiquement, nous proposons d'utiliser la méthode des agents logiciels pour concevoir et développer une telle architecture. Des agents automatiques ou semi-

automatiques collaborent avec les concepteurs humains, et ils peuvent coopérer les uns avec les autres. Pour ces raisons, nous avons choisi un modèle de tâches à plat plutôt que hiérarchique.

Identification de certains agents et adaptation aux tâches utilisateur Dans le cas précis des contraintes et des critères, il est possible d'imaginer plusieurs agents. Ils s'occuperont de vérifier les inconsistances dans l'espace de conception, de générer des solutions partielles et complètes, de présenter des exemples de conception, d'aider l'utilisateur à visualiser un ensemble de bonnes solutions, de détecter des conflits entre contraintes, et de proposer des compromis. Ces agents doivent être semi-automatiques ; ils ne seront appelés que sur demande de l'utilisateur. Ces agents pourraient devenir davantage autonomes à partir du moment où une certaine confiance a été établie entre l'utilisateur et l'agent.

Les compétences et les connaissances étant distribuées sous différentes entités, toutes les familles d'utilisateur trouveront de l'assistance. Par exemple, nous avons imaginé pour le CL trois types d'utilisateurs (dans notre cas, situation N° 3 de la figure 1) :

- *Un utilisateur précis* qui possède des contraintes topographiques; des meubles ainsi que le plan de sa pièce. L'utilisation de l'aide à la satisfaction de contraintes sera particulièrement importante pour lui.

- *Un utilisateur flou* qui exprime des critères flous tels que la clarté, l'ancienneté, le style, etc. L'utilisation de l'aide à l'optimisation multi-critères lui sera d'une grande utilité.
- *Un utilisateur indécis* qui n'a aucune idée a priori de son aménagement. L'utilisation d'un catalogue de solutions sera alors utile pour l'aider à définir ses critères et contraintes.

Contraintes et critères seront définis à un niveau global et seront partagés par tous les sous-systèmes intelligents, correspondants aux tâches de la conception. Un système en tâches hiérarchiques ne pose pas de problèmes supplémentaires pour la prise en compte des critères et des contraintes. Néanmoins, il contredit le déroulement "ouvert" d'une tâche de conception.

Les agents qui nous intéressent tout particulièrement concernent la recherche de solutions par résolution de contraintes et la recherche de compromis par optimisation multi-critères. Notons, de plus, que la résolution des conflits peut être une étape particulièrement importante dans la recherche de la meilleure solution.

Un agent essentiel : Le système de résolution de contraintes En considérant le système de résolution de contraintes comme un agent computationnel autonome des autres fonctionnalités du système, l'utilisateur pourra l'appeler lorsqu'il le souhaite.

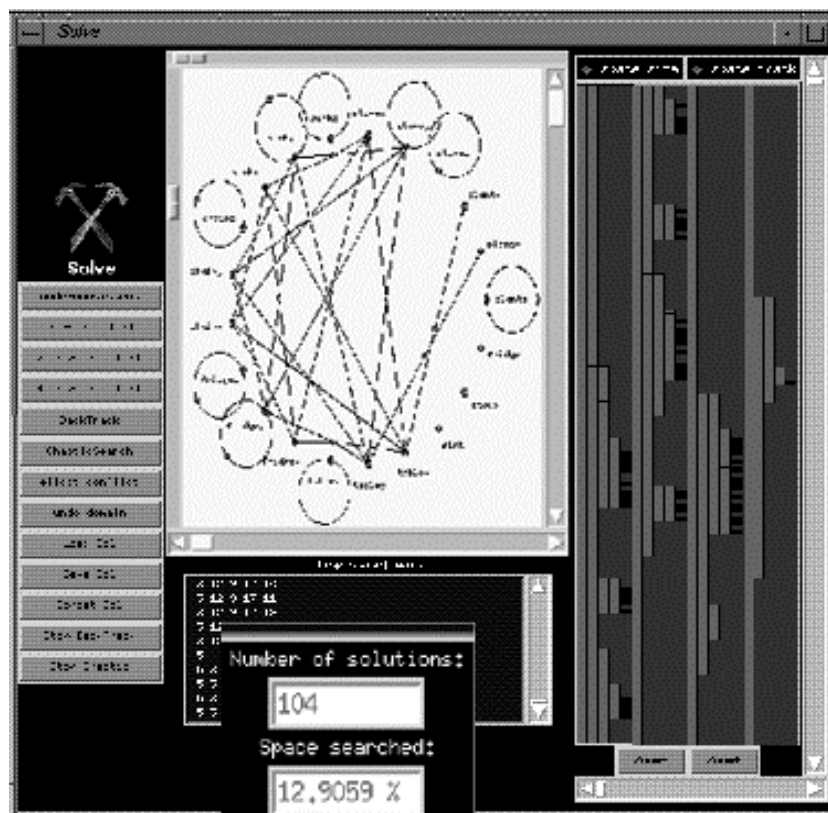


Figure 2 : L'agent SOLVE, pour une résolution interactive des contraintes.

Il serait dommage de brider la créativité de l'utilisateur avec un système vérifiant que toutes les contraintes sont satisfaites à tous les moments de la conception. Pour laisser l'utilisateur libre d'explorer l'espace de conception, nous avons imaginé que certains agents pourraient se charger de refléter le cheminement du concepteur, en conservant un historique de ses décisions et de ses modifications de la définition du problème. Les cartes cognitives sont souvent très utiles afin de supporter les humains dans l'exploration d'espaces inconnus. Parce qu'il est facile pour un concepteur de s'écarter de ses objectifs, les cartes cognitives peuvent servir de prothèse à leur système attentionnel de façon à ce qu'ils puissent le remarquer. De plus, on peut imaginer que le système de résolution de contraintes propose une visualisation de l'espace de recherche ainsi que différents algorithmes de recherche afin de donner un plus grand contrôle à l'utilisateur. Le concepteur pourra ainsi interactivement explorer l'espace de solutions ; la machine augmentant ainsi les capacités computationnelles du concepteur. La figure 3 montre l'interface d'un système de résolution de contraintes interactif. Plusieurs algorithmes de recherche sont proposés; une recherche peut-être interrompue ou restreinte à un espace particulier. La partie droite de la figure 3 montre une visualisation de la recherche en cours. Les rectangles foncés sont les valeurs interdites par les contraintes. L'utilisateur peut sélectionner à la souris une partie de l'espace sur laquelle il a l'intuition qu'une recherche s'impose. Ainsi le concepteur peut choisir à l'aveuglette un espace de recherche qui lui semble intéressant, faculté essentielle à la base de nombreuses découvertes.

Les critères : trouver la "bonne" solution parmi ... une infinité ? Lorsqu'un problème de conception possède de nombreuses solutions, il est important que ces solutions puissent être comparées. Souvent, les solutions sont en compétition ; elles font partie d'un ensemble appelé Pareto optimal. Des solutions sont bonnes suivant certains aspects mais pas pour tous et ne peuvent pas donc dominer les autres. De la créativité est nécessaire pour découvrir de nouveaux critères pour bousculer l'ensemble des solutions Pareto optimales et pour les réévaluer afin de trouver des compromis. "Voir les choses sous un autre angle", en changeant ses critères ou en visualisant dans un contexte différent, c'est ce que l'on fait souvent dans la vie de tous les jours pour prendre des décisions.

Le système d'aide à l'évaluation des solutions devrait permettre de visualiser les solutions selon différents critères et ainsi de trouver des compromis dans le cas où ceux-ci sont antagonistes.

La figure 3 présente un scénario où l'utilisateur peut évaluer graphiquement des cuisines, qui satisfont ses contraintes, en fonction de différents critères, qu'il a défini lui-même, tels que le prix et l'espace disponible. La machine permet ainsi d'augmenter les facultés perceptuelles de l'utilisateur avec différentes visualisations. A cet instant du scénario, l'utilisateur peut se rendre compte que certaines mauvaises solutions

n'ont été interdites par aucune contrainte. Il va donc raffiner ses contraintes.

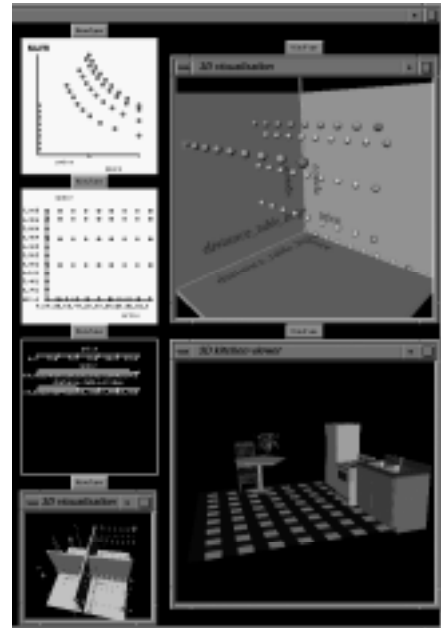


Figure 3: Un exemple de système d'aide à l'évaluation de solutions.

Il peut aussi constater que les solutions optimales ne lui conviennent pas et va donc ajouter des critères auxquels il n'avait pas pensé auparavant

Conclusion La prise en compte des contraintes et des critères dans un système d'aide à la conception ne semble pas devoir se résumer à un simple système de résolution de contraintes. Le processus de conception consiste justement à définir contraintes et critères, ce qui ne se fait pas d'un seul coup et qui n'a rien d'une tâche automatique. Les critères tout spécialement sont les seuls outils d'évaluation du concepteur. Cependant, même si ses objectifs sont clairs a priori, il lui faut consacrer beaucoup d'efforts pour trouver les critères leurs correspondant.

S'il semble naturel à la communauté IHM de soutenir les phases de définition, d'exploration et de décision liées à la conception, l'étude du traitement des contraintes et des critères devrait être aussi l'une de leur préoccupation afin de trouver les objets d'interaction communs entre l'homme et la machine.

ANALYSE DE L'UTILISATEUR ET DE SON ACTIVITÉ

Dans le cadre de cet atelier, peu des participants travaillent dans le domaine de l'analyse de l'utilisateur et de son activité. Notre travail, ici s'est limité à la proposition d'une méthode par [1] et [2] à caractère sémiotique [3], [16], [18]. Celle-ci a été mise au point pour analyser des enregistrements vidéo de situations du second type (des personnes jouant les rôles du CL et du CU dialoguent devant l'écran d'un logiciel de synthèse d'image que manipule le CU). La sémiotique dispose de modèles narratifs et de principes de segmentation du

perçu qui ont permis de découper le récit de l'activité, présenté par la bande vidéo, en fragments de plus en plus fins allant jusqu'aux expressions multimodales conjoignant parole, geste et image. Le schéma narratif de base précise qu'un faire implique un vouloir, un savoir et un pouvoir faire, et que le faire comme l'être qui en résulte sont soumis à une évaluation.

L'enregistrement vidéo manifeste les différentes composantes du schéma sur un mode itératif. Le segment narratif de base enchaîne : choix de l'élément de cuisine et de l'emplacement - faire graphique - évaluation de l'état de la scène résultant. Le segment graphique de base, correspondant au faire graphique, se décompose en : apparition de l'élément — rotation (facultative) — placement d'une copie de l'élément — disparition de l'original. Les expressions multimodales apparaissent comme des "instruments sémiotiques" contribuant à la construction des vouloir, des savoir et des pouvoir faire, ainsi que des faire et des évaluations.

MODELES DE TÂCHES ET DE DIALOGUE

Modèle séquentiel versus stratégie opportuniste

Le groupe de travail a soulevé le problème du modèle de tâche adapté pour les systèmes de CAO. Est-il préférable de disposer de modèles séquentiels, hiérarchiques ? Dans le cadre d'application de CAO, certains préfèrent parler de stratégie opportuniste. Nous allons dans un premier temps préciser la tâche de conception en termes de but et de résolution de problèmes.

Les modèles séquentiels de tâche font apparaître explicitement des suites de sous-buts (ou d'états intermédiaires) qui une fois atteintes créent des contextes qui orientent et contraignent les opérations possibles.

Cette vision de résolution de tâche par plans est conforme à l'approche résolution de problème—d'ailleurs, ce point est à rattacher au paragraphe précédent relatif au rôle des contraintes et des critères dans le processus de conception—. où chaque sous-but (ou état) correspond à une résolution partielle du problème. La conception ne peut pas être vue sous cet angle dans la mesure où il n'existe pas une solution cible entièrement préspecifiée mais un état considéré comme final par un acteur humain. Cet état est atteint par une stratégie opportuniste et des critères d'évaluation en partie flous des états intermédiaires. Toutes les opérations possibles peuvent être envisagées (et de préférence rendues possibles par l'interface) à n'importe quel moment du processus de conception incrémentale. D'où le modèle de tâche qui propose un vrac de ressources permettant d'appliquer la **stratégie opportuniste** propre à la **conception incrémentale**. Il existe bien sûr des raffinements à ce modèle mais il faut alors envisager une étude plus spécifique de cette question intégrant les fonctionnalités propres d'un système interactif pour ce cas d'application.

Il reste que la séquentialité est néanmoins dominante dans un processus qui se déroule dans le temps. Si un tel

processus est qualifié d'incrémental c'est parce qu'il fonctionne sur une accumulation d'essais et d'erreurs, de solutions intermédiaires incomplètes et insatisfaisantes. Cette accumulation constitue la mémoire du concepteur (voire de la machine si des moyens de sauver des étapes de conception sont possibles) et joue un rôle important d'une part dans la manière dont se développe le processus et d'autre part dans la manière dont le concepteur va s'exprimer (faire références à des états antérieurs, reproduire tout ou partie de ces états etc.). Le schéma séquentiel proposée explicite cette dimension temporelle du processus.

Toute tâche de conception, que ce soit celle d'un programme informatique, d'une interface, d'un article scientifique ou d'une cuisine, comporte une phase d'initialisation, une phase qui est le cœur du processus de conception et enfin une phase de finalisation. Cette vision séquentielle en trois phases successives est d'une grande généralité et très pauvre du point de vue de la description de la réalisation de la tâche, on ne doit donc pas voir là un modèle de tâche mais un schéma général. En quoi est-il utile ? Il fait apparaître explicitement des phases qui correspondent à des tâches cognitives qui se différencient par la nature des informations exprimées et prises en compte, la manière de les exprimer et les objectifs à atteindre. Les bornes de ces phases ne sont pas faciles à définir, ce qui a été proposé au cours de l'atelier est de marquer la fin de la phase d'initialisation par la visualisation d'une maquette zéro dont la forme d'expression est analogue à la maquette de cuisine sur laquelle le client se prononcera en dernier lieu. Le marquage de la fin de la phase correspondant au cœur de la conception pourrait être l'acceptation de la cuisine proposée, la phase de finalisation visant à régler des détails du type fiche client et devis précis. Rien n'empêche de passer de la finalisation au cœur de la conception (des détails peuvent avoir été oublié) mais il ne peut pas y avoir de retour à la phase d'initialisation du point de vue cognitif des acteurs humains même si le programme est réinitialisé sur d'autres informations d'entrée (cf. l'effet de mémoire décrit plus haut).

Dans le cadre de cet atelier, deux modèles de tâche empruntés au dialogue homme-machine (DHM) fondé sur une approche langagière sont décrits.

Modèles de tâche et de dialogue : explicites et implicites

"Analyse de tâches", "modèle de tâche", "activité" sont des termes issus de l'ergonomie qui se sont propagés dans de nombreux domaines, du génie logiciel, de l'intelligence artificielle, de la robotique, pour aboutir au domaine du dialogue homme-machine (DHM). Si dans leurs utilisations dans ces divers domaines ces termes ne recouvrent pas toujours exactement les mêmes notions, l'objectif de base en DHM et en conception d'interfaces est d'analyser, de représenter et de contrôler la suite des actes d'un utilisateur.

La suite d'actes, réellement effectuée par l'utilisateur pour résoudre un problème, est appelée activité. Elle

n'est de fait explicable qu'a posteriori —une fois le but atteint —et vis-à-vis de l'objectif visé. L'activité est l'ensemble indifférencié des actes manifestés par un utilisateur : il en résulte que ces actes peuvent avoir ou non un rapport direct avec l'objectif (dégager un espace de travail sur l'écran ne fait pas vraiment partie de la tâche d'écriture d'un texte par exemple, cela est seulement nécessité par l'encombrement momentané de l'écran). La tâche prend sa signification par rapport aux buts que s'est assignés l'utilisateur. Comprendre l'activité revient à donner un sens aux actes de l'usager compte-tenu de ses intentions (supposées ou connues) vis-à-vis de la tâche qu'il a à accomplir.

Une tâche se définit comme la réalisation d'un but dans un contexte et selon une procédure dont la représentation et la précision varient en fonction des objectifs dirigeant l'analyse. C'est en effet du ressort de l'analyste de dire ce qu'il entend par tâche. Souvent il adopte un point de vue fonctionnaliste : il représente plus ou moins abstraitement les actes potentiels d'un utilisateur, leurs effets et leurs conditions de réussite, leurs contraintes. Le degré de granularité des actes est très variable, leur décomposition ou leur structuration est souvent arbitraire. La complexité des tâches dépend bien sûr des domaines d'application : elle va de la routine (contrôle de processus) à l'innovation (problèmes de conception) en passant par tous les stades de complexité intermédiaires. Dans le premier cas on conçoit bien qu'elle soit fortement structurée et que l'activité soit planifiable a priori, dans l'autre cas elle ne l'est pas du fait de l'imprévisibilité même des processus de création. Il est alors tentant pour le concepteur du dialogue de représenter des formes de comportements types pour canaliser et interpréter l'activité de l'utilisateur. Remarquons cependant que dans de nombreuses situations commandées, l'utilisateur ne suit pas nécessairement l'ordonnement prévu dans le modèle de tâche — même pour des tâches à risque et/ou très contraintes [25]. Le modèle de tâche n'est donc utile à la machine que pour planifier son propre comportement et pour tenter de se représenter le sens des actes d'un utilisateur.

Un modèle de tâche doit donc être entendu à divers sens : représentation des enchaînements des actions de la machine, représentation structurée des fonctions du logiciel, suite prototypique d'actes conduisant à un but, ensemble des connaissances servant à organiser une activité, etc.

On peut distinguer deux grands types de « modèles de tâche » :

- les modèles explicites, dans lesquels la tâche guide le dialogue
- et les modèles implicites, dans lesquels l'activité guide le dialogue (ce qui donne des stratégies moins directives).

Les modèles de tâche explicites Un modèle de tâche explicite décrit de manière explicite la succession des actions (corps, effets, conditions, ordonnancement, etc.) possibles conduisant à un but donné. Classiquement les tâches sont hiérarchisées (arbre ET/OU, réseau ATN, réseau de Pétri, etc.) en plans et/ou sous-plans ou sous-tâches, scénarios, etc., jusqu'aux scripts qui instancient les actions élémentaires. Ce mode de représentation est statique : il décrit la combinatoire des actions qu'il est théoriquement possible d'enchaîner pour exécuter une tâche avec succès.

Un tel modèle de tâche permet de traiter le dialogue et la tâche comme des problèmes de planification a priori. Pour des tâches de conception ces modèles ne peuvent convenir puisque la tâche de conception ne peut être planifiée à l'avance. Un autre défaut de ces modèles est en outre de considérer que le dialogue est conditionné par la tâche voire guidé par la tâche. Cela n'est vrai que pour des dialogues opératifs.

Les modèles de tâche implicites Un modèle de tâche implicite ne décrit pas la succession des actions mais seulement le but à atteindre et des moyens pour l'atteindre (le cheminement n'est pas explicité, il sera inféré en relation avec la situation).

Durant la dernière décennie on a cherché à comprendre certains comportements humains placés en résolution de problèmes. La plupart d'entre eux utilisent non pas des plans prédéfinis mais raisonnent par analogie en s'appuyant sur des situations et des savoirs-faires connus, ou raisonnent par généralisation, abduction, etc. Les tâches de conception sont à cet égard les plus caractéristiques : les concepteurs construisent leurs buts au fur et à mesure de l'évolution de leur tâche, de manière opportuniste en ayant seulement en arrière-plan un objectif général. Cela conduit à représenter le modèle de tâche par :

- l'objectif général et les buts,
- des contraintes pour atteindre les buts
- et des algorithmes de "mutation" de scripts-prototypiques permettant d'ajuster ou de modifier des scripts ayant déjà été utilisés dans d'anciennes situations.

Le laboratoire CLIPS-IMAG a choisi de travailler dans ces directions. Dans un modèle de tâche implicite l'effort ne porte non plus sur la représentation des données mais sur la puissance des processus d'inférence. Il y a dans cette perspective deux problèmes majeurs :

- la gestion des buts,
- l'apprentissage des savoirs-faires.

Le gestionnaire des buts opère sur une liste dont il peut modifier l'ordre en fonction de l'évolution du dialogue, des circonstances, de l'urgence de la situation,

des états mentaux de l'utilisateur, etc. On suppose qu'à un instant donné on se trouve dans un échange (une suite de tours de parole pendant lesquels un but est maintenu). On ne sort d'un échange que lorsque le but est satisfait ou que par un abandon. Pour gérer les buts (et donc planifier les actions de la machine) on dispose d'un arbre de buts et de marqueurs indiquant l'état des buts. Les fonctions élémentaires de gestion sont l'activation d'un nouveau but, l'atteinte, la satisfaction, la mise en attente, la réparation, la satisfaction, la mise en attente, la réparation, le déplacement d'un but et la réduction à un ou plusieurs sous-but.

L'objectif général est de donner à la machine des capacités d'adaptation à la tâche. Ces capacités nécessitent de mettre en œuvre des processus d'apprentissage car il n'est pas envisageable de prévoir toutes les situations d'usage ni tous les types d'utilisateurs a priori. La machine doit donc doublement s'adapter :

- d'une part, elle doit acquérir les concepts manipulés à travers le langage et qui sont souvent "naturels" (donc implicites) pour l'utilisateur humain,
- d'autre part elle doit apprendre des plans d'action dans le contexte d'usage de l'utilisateur et de manière suffisamment générique pour être réutilisables.

Comme dans le dialogue humain, il est opportun de profiter du dialogue non seulement pour obtenir des renseignements, échanger des points de vue, coordonner les actions, etc., mais aussi pour apprendre. L'idée est donc de fonder le modèle de dialogue sur la notion d'apprentissage des savoirs et des savoir-faires. Cela conduit notamment à la recherche d'un modèle adéquat de représentation des connaissances apte à faciliter l'apprentissage incrémental, l'élaboration de mécanismes de raisonnement et la construction de plans. Pour être capable d'un tel dialogue, la machine doit avoir des capacités qui lui permettent de coordonner les processus actionnels en fonction des buts de l'usager [8].

La suite présente, sans détailler, certaines considérations relatives à l'apprentissage dans un modèle de dialogue proposé pour affronter des situations de conception. Dans ce type de dialogue, le plan pour exécuter la tâche (ou pour résoudre le problème) est construit au fur et à mesure de l'avancée du dialogue et par le dialogue. Après que le but ait été atteint et validé par l'utilisateur, la machine connaît les actions et peut réitérer la tâche si nécessaire ou la généraliser à d'autres usages. Dans des situations ultérieures et à partir de cette connaissance apprise, la machine a aussi la possibilité d'inférer les intentions de l'utilisateur pour coopérer à (ou anticiper sur) la résolution de la tâche en cours — puisqu'il lui suffit de reconnaître les séquences significatives de la tâche en question.

Par définition la conception est une activité créatrice. Malgré tout, on s'aperçoit — chez les architectes par exemple — que l'acte de conception passe souvent par une réutilisation de plans architecturaux anciens, simplement réorganisés ou réagencés. Au-delà de ce simple exemple notons également l'importance de l'apprentissage par l'action en psychologie du comportement [22].

Les problèmes soulevés à propos de l'apprentissage de la tâche sont :

- l'acquisition d'un nouveau plan d'action, pertinent pour la tâche en question,
- sa caractérisation par rapport aux autres tâches,
- l'assignation d'un but dans le contexte du dialogue.

Puis, dans un deuxième temps, pour l'assistance à l'utilisateur, les problèmes sont :

- la reconnaissance du but poursuivi à partir des actions observées, pour inférer la logique d'action générale de l'utilisateur et identifier son plan d'action,
- l'activation pertinente des stratégies d'assistance à l'utilisateur dans le cours du dialogue.

L'apprentissage se fait pendant une phase d'observation suivie par une phase de généralisation confirmée par l'utilisateur. Le contrôle de l'activité se fait en proposant par anticipation ou de manière coopérative des schémas appris. Au départ, on suppose que l'on dispose d'actions élémentaires bien. Puis, après que l'utilisateur a posé un but, il s'agit pour la machine d'observer, d'ordonner et d'associer cette séquence d'actions au but posé. Nous sommes donc ici dans un contexte maître-apprenant où le maître est l'utilisateur et l'apprenant est la machine :

- pendant la phase d'observation la machine enregistre la série d'actions montrées par l'utilisateur,
- à partir de l'enregistrement, la machine tente de distinguer des relations entre les actions, par exemple, la succession dans le temps ou la causalité ; pour établir un ordre, probablement partiel, entre les actions, et
- dans la phase d'association, on attache le plan obtenu au but poursuivi par l'utilisateur.

À la fin de cette étape, la machine est capable de recommencer la tâche à partir d'une demande de l'utilisateur. Mais l'apprentissage ne s'arrête pas ici. La nouvelle tâche apprise doit être généralisée, spécialisée ou intégrée comme une sous-tâche à une tâche plus complexe. Cette assimilation permet la formation d'une

abstraction liée au but posé par l'utilisateur. De cette façon, la connaissance acquise évolue pour se rapprocher au concept réel détenu par l'utilisateur.

Une difficulté majeure est la manière de représenter les connaissances pour supporter cet apprentissage, car on fait l'hypothèse que les connaissances acquises par la machine sont incomplètes et dynamiques et qu'elles doivent rendre explicites les différents rapports entre les actions. Les idées ci-après sont basées sur le travail de Y. Shohlam qui définit d'une manière générale une proposition temporelle $\langle i, p \rangle$ laquelle associe une assertion logique p à un intervalle de temps i . La syntaxe et la sémantique de cette logique sont définies dans [9]. Pour la formation d'un concept, notre travail s'appuie sur le paradigme inductif en utilisant des stratégies d'apprentissage de "concepts par l'exemple". L'idée principale est l'adaptation de la description d'une tâche à chaque fois que l'utilisateur donne une instance de cette tâche. Dès que l'on apprend une nouvelle tâche, le concept créé est la première approximation de la description de celle-là. Lorsque l'utilisateur présente une nouvelle instance de la tâche, un processus de généralisation adapte l'approximation actuelle du concept pour inclure les caractéristiques propres de l'instance montrée. Le résultat de la généralisation, entre l'approximation courante et l'instance actuelle, est une nouvelle approximation plus complète. Cependant, dès la première approximation, bien qu'inexacte et/ou incomplète, on peut l'utiliser pour reproduire la tâche.

Dans notre cas, l'apprentissage est caractérisée par :

- la formation d'un concept, il est (i) empirique — au départ, on ne connaît rien sur la tâche —, et il (ii) procède de façon incrémentale — la description d'une tâche est ajustée à chaque occurrence d'une nouvelle instance de cette tâche.
- les tâches sont classifiées avec l'aide et sous le contrôle de l'utilisateur. L'utilisateur guide la formation de nouveaux concepts, ainsi que l'affinage de concepts déjà appris.
- la description d'une tâche doit permettre sa reconnaissance ; dans une situation ultérieure, on tente d'inférer la tâche à partir des actions observées.

La reconnaissance d'intentions peut être vue comme le problème inverse de l'apprentissage : les notions résultantes de l'apprentissage dirigent, pendant le déroulement du dialogue, la reconnaissance du but poursuivi par l'utilisateur, c'est-à-dire, qu'à partir des concepts appris, on tente de retrouver l'intention de l'utilisateur. Ainsi, la machine sera capable de coopérer lorsqu'une situation, analogue à une autre déjà vécue, est rencontrée. Dans ce cas-là, après la confirmation de l'utilisateur et éventuellement un sous-dialogue de particularisation de la tâche, le plan pourra être exécuté. De cette manière, en s'appuyant sur une stratégie dirigée

par les intentions, la machine établit un dialogue qui converge vers la réalisation de la tâche.

La recherche des intentions utilise un processus de raisonnement pour déduire le but de l'utilisateur à travers la succession d'actions observées. Comme nous l'avons évoqué, l'action est l'élément principal. Le dessin d'une maison et d'un bateau sont différents par les actions réalisées non par les formes perçues. Pour reconnaître l'intention de l'utilisateur on doit enregistrer et transformer les événements observés ; et ensuite vérifier s'ils font partie d'un concept connu.

Le système de reconnaissance tente donc d'établir un lien entre une séquence répertoriée et la séquence d'actions observées. Cette opération est réalisée de manière incrémentale. Au fur et à mesure de l'occurrence des événements une description récente est construite. Lorsque cette description récente est appariée avec un sous-ensemble d'actions d'un concept connu, la machine peut donc continuer la tâche après la validation de l'utilisateur. Dans le cas d'ambiguïté, c'est-à-dire, quand le processus de reconnaissance obtient plus d'une hypothèse, on attend jusqu'à ce que la description courante soit suffisamment complète pour tomber sur une seule possibilité.

INTERACTION MULTIMODALE ET REFERENCE

Toutes les modalités d'interaction —gestuelle, langagière, ... ont été brièvement discutés dans le cadre de cet atelier. Pour cet article de synthèse, il nous a semblé important de focaliser sur les points suivants :

- l'interaction au stylo dans le cadre de l'aménagement de cuisines,
- les moyens d'expression utilisés par les différents acteurs,
- les divers études et approches de la référence intermodale et des déictiques.

Interaction au stylo

Le stylo permet une expression directe des intentions de l'utilisateur, sans recourt aux conventions et aux langages imposés par les normes du domaine et les interfaces de D/CAO. Un client peut ainsi participer à la description graphique de sa cuisine en la dessinant comme il pourrait le faire sur une feuille de papier. C'est ce qui a motivé la réalisation de Production Assistée de Plan de Pièces d'Habitation (PAPPH) [11, 12].

Ce mode d'interaction est particulièrement pertinent pour les phases précoces de la conception graphique. Cette phase correspond à l'esquisse de la scène pour la mise en place les éléments sans qu'il soit encore nécessaire de les spécifier complètement. Les interfaces à stylo n'ont pas pour but de remplacer les logiciels de D/CAO existant, plus adaptés à visualiser le projet dans des phases avancées de sa conception. Dans le cas des cuisines, ces logiciels permettent de choisir des styles de

meubles, des revêtements, des matériaux, des couleurs. C'est donc comme frontal d'un système plus conventionnel de D/CAO qu'une interface comme PAPPH doit être considérée (la visualisation d'une image 3D de la scène 2D illustre le passage à d'autres formes de représentation des données).

Techniques d'interaction

Les moyens d'expression utilisés par les différents acteurs sont fonction de la nature du message à communiquer (relation entre la forme et le fond), des moyens disponibles et de leur préférences.

On peut distinguer trois types d'intervention de CL dans le processus de conception :

- CL informe CU en parlant et en dessinant (sur un papier ou sur un papier électronique),
- CL réagit à une proposition visuelle en désignant une image, ou des éléments dans une image, qui peut être affichée sur l'écran ou sur un catalogue, et en parlant,
- CL agit sur M (Machine) en modifiant l'image affichée.

On peut caractériser les images affichées par M en fonction des dimensions décrites dans [13]. Ces dimensions font apparaître les relations entre les interactions possibles et ce qu'elles imposent du point de vue des connaissances de la machine, des procédures de traitement et du couplage entre interface et noyau applicatif.

Les images affichées ont une fonction de référence si l'on considère le dialogue entre CL et CU, elles ont aussi pour fonction de visualiser l'état d'avancement du projet d'aménagement de cuisine mené par CU et CL ou des bases d'images préenregistrées de type éléments du mobilier, catalogue ou prototypes de cuisines.

Les images affichées doivent pouvoir être modifiées si elles correspondent à la cuisine en cours de conception. La forme de malléabilité de ces images est fonction du mode d'interaction mais aussi des capacités d'initiative dont dispose la machine. Les images ne seront modifiables que si elles sont réceptives. Directement réceptives comme dans PAPPH où U (Utilisateur au sens générique) intervient directement sur l'image en dessinant au stylo, ou indirectement réceptives si l'image est modifiée suite à l'exécution de commandes d'un logiciel de CAO qui sont émises par U. L'adaptativité sera possible si M dispose de connaissances (valeurs par défaut, procédures de calcul etc.) qui lui permettent de prendre des initiatives pour compléter des commandes incomplètes ou pour actualiser automatiquement l'image en fonction de nouvelles informations qu'elle reçoit, c'est le cas dans PAPPH où un changement des dimensions de la pièce provoque une mise à l'échelle automatique des éléments qui la composent. La possibilité de produire

une image sans avoir à fournir l'ensemble des informations nécessaires à l'affichage s'oppose aux formes de commandes classiques en CAO. Cette possibilité implique des procédures qui combinent des connaissances du domaine aux informations exprimées par U pour produire une image, la difficulté à mettre au point ces procédures de complétion de commandes ne doit pas être vue comme une impossibilité à partager la tâche entre U et M. Le dialogue peut faciliter ce partage des tâches en permettant à la M de poser des questions ou de proposer plusieurs solutions en cas de trop forte ambiguïté. Des propositions sur ce type de question ont été faites dans [14]. L'avantage de pouvoir produire des images incomplètement spécifiées est très sensible dans un processus de conception où par nature les états intermédiaires relèvent plus d'une intention schématique, approximative que d'une intention précise du concepteur. D'autre part, il est important pour le concepteur de disposer rapidement d'une image, d'un support visuel, pour aider sa démarche.

L'image peut devenir réactive si les initiatives de U et de M se combinent lors d'une interaction ayant l'image comme support (voir aussi à ce sujet, la prise en compte des contraintes dans les IHM). Par exemple, dans PAPPH, U prend l'initiative de déplacer un élément en agissant sur sa représentation graphique. Cette représentation est associée à un comportement fonction des connaissances du domaine ce qui amène U à expérimenter une réaction de M qui peut empêcher le déplacement (les portes ne peuvent pas être décollées des murs, les éléments solides ne peuvent pas se pénétrer). Les connaissances du domaine qui permettent ces comportements portent sur les éléments de la scène et les relations spatiales. Chaque type d'élément est caractérisé par sa taille minimale et maximale et la manière dont sa taille peut varier. Pour certains éléments, la variation de taille est continue alors que pour ceux qui ne peuvent avoir que des tailles standards, la variation est discrète. Les éléments ne peuvent pas se superposer et les portes restent attachées aux murs qui les supportent. Ces connaissances sont utilisées lors de l'exécution des commandes de l'utilisateur pour contraindre les retours. On a là une interaction sémantique qui associe des comportements aux objets de la scène qui sont fonction de leur nature et du contexte.

L'image affichée peut être le résultat d'une production directe si U dessine son projet au stylo, à la souris ou en plaçant des icônes. Elle peut être produite par M si elle est le résultat du traitement des informations fournies par U on aura une production constructive (l'image est construite à partir de commandes CAO ou de l'analyse de tracés à main levée). Pour les images stockées dans des bases (catalogues, éléments de cuisine), M affiche le contenu des fichiers, c'est une production restitutive. Une production analytique pouvant répondre à des commandes du type : "je veux l'évier et la plaque de cuisson de la cuisine X mais la hotte de la cuisine Y" implique que la machine comprend la commande exprimée en langue

naturelle mais aussi puisse analyser les images pour accéder à leur contenu.

L'interaction multimodale : Étude des Références et des Déictiques

Le groupe a abordé la multimodalité et le problème de l'étude des références intermodales —résolution de la référence intermodale—dans deux situations d'interaction :

- de type Magicien d'Oz
- et de systèmes interactifs déjà réalisés.

Expérimentations Magicien d'Oz

Expérience du CRIN [9, 19,23] L'expérience dont est extraite les résultats qui suivent a été réalisée grâce au protocole du Magicien d'Oz. Les sujets avaient à modifier l'arrangement d'une pièce d'un appartement et à y installer de nouveaux meubles. Tout se passe en 2D et les gestes étudiés sont également 2D.

Nous nous intéressons ici à l'utilisation des démonstratifs et des déictiques. En effet, lors de l'expérience, il est apparu que le geste pouvait être subordonné à la parole par l'usage de l'adjectif "ce" du pronom relatif "celui-ci" ou de l'adverbe "ici".

Les adjectifs démonstratifs "ce, cet, cette" peuvent avoir deux usages :

- anaphorique, et dans ce cas ils font référence à la dernière action du système ou à un meuble déjà introduit,
- associé à un geste de désignation. Dans ce cas, ils ne sont pas spécialement associés à un geste de pointage simple (comme c'est le cas le plus souvent en communication Homme-Homme). En effet, les démonstratifs sont souvent associés à un geste continu décrivant l'ensemble du déplacement.

Le pronom démonstratif "celui-la, celui-ci, celle-ci" reprend le plus souvent une catégorie déjà introduite dans le discours mais change ou précise l'instance de cette catégorie par un geste de désignation. Comme pour les démonstratifs, il est associé, soit à un geste discret, soit à un geste continu.

Enfin, les adverbes "ici", "là" sont également employés lors de l'expérience. Le déictique "ici" est un élément important des commandes de déplacement. Tous les énoncés contenant "ici" sont multimodaux et le geste associé est le plus souvent continu. "là" est plus rare et permet de faire référence à la situation et non à une position dans le plan.

Il faut remarquer que la plupart des déictiques sont utilisés par des sujets étant familier de l'informatique et qui ont donc déjà développés leur propre représentation de la machine. Les sujets plus novices utilisent moins de déictiques et sont plus influencés par la manipulation

directe; ce qui peut expliquer leur combinaison avec des gestes continus et non discrets comme on le fait en communication Homme-Homme.

Expérience de [1] Les travaux décrits ici concernent la référence déictique multimodale image-parole-geste correspondant à la situation (N° 3 de la figure 1) où le cuisiniste manipule un logiciel de synthèse d'image. Les références déictiques interviennent rarement dans les phases de faire graphique, et fréquemment lors de la construction d'unvouloir (choix de l'objet à placer et de l'emplacement, ou corrections de choix réalisés) et de l'évaluation d'un état imaginé ou résultant d'un faire. L'image visualisée sert de référence à des expressions linguistiques complétées par ou en redondance avec des gestes. Les cas de multimodalité alternée où parole et geste se partagent les composants d'un énoncé (objets, leurs attributs, emplacements, actions) sont beaucoup moins fréquents que les cas de multimodalité synergiques. Les gestes de désignation et de manipulation ont diverses fonctions. Ils spécifient de quel exemplaire d'une classe ou d'une collection, on parle ("ici", "celui-là", "comme ça", "ce mur", "le placard", ...). Ils visualisent des aspects de sous-espaces virtuels - limites d'objets, emprises au sol ou sur un mur, dimensions (longueur, largeur, hauteur) - ainsi que des repères de composition architecturale (alignement par exemple). Les références déictiques multimodales sont sémantisées en fonction du domaine considéré. Elles se rapportent aux objets du domaine (éléments de mobilier, composants de la pièce), et à des aspects de ces objets et aux zones de l'espace contigus à ces objets pertinents dans le domaine. Elles traitent également des zones singulières importantes dans le domaine : coins d'une pièce, intervalles entre des meubles, espaces au bout d'une rangée de meubles, etc. Elles s'intègrent par ailleurs à des plages de focalisation sur un ou plusieurs sous-espaces, qui coïncident avec une suite de plusieurs énoncés. Ces plages de focalisation d'ordre perceptivo-cognitif sont liées à la tâche et traduisent l'attention portée par les interlocuteurs à un problème déterminé. Elles se rapportent aux objets du domaine (éléments de mobilier, composants de la pièce), et à des aspects de ces objets et aux zones de l'espace contigus à ces objets pertinents dans le domaine. Elles traitent également des zones singulières importantes dans le domaine : coins d'une pièce, intervalles entre des meubles, espaces au bout d'une rangée de meubles, etc. Elles s'intègrent par ailleurs à des plages de focalisation sur un ou plusieurs sous-espaces, qui coïncident avec une suite de plusieurs énoncés.

Ces plages de focalisation d'ordre perceptivo-cognitif sont liées à la tâche et traduisent l'attention portée par les interlocuteurs à un problème déterminé.

Systèmes multimodaux déjà réalisés Coréférences et déictiques dans LIMSI-Draw

Les coréférences utilisées dans LIMSI-Draw [5] sont essentiellement du type « entrée d'une donnée + référence à un argument ». Dans l'exemple " **mets (ça + pointage objet) (ici + pointage position) "** le mot " ça " faire référence à l'argument objet de la

commande déplacer, alors que le mot "ici" fait référence à l'argument "position" de la même commande. Les pointages de l'objet et de la nouvelle position sont des entrées de valeur. Celles-ci peuvent être effectuées aussi bien à travers la souris que l'écran tactile.

Ce type de coréférences s'est avéré particulièrement utile dans les trois cas suivants :

- **Correction** Si l'utilisateur se rend compte qu'il vient d'entrer une mauvaise donnée, il a toujours la possibilité de la rectifier. Il lui suffit pour cela d'entrer la nouvelle donnée en l'accompagnant d'une référence à l'argument auquel elle doit être affectée. L'ancienne donnée sera alors remplacée par la nouvelle. La correction peut être répétée autant de fois que l'utilisateur le désire.
Reprenons l'exemple de la commande de déplacement d'objet. Si l'utilisateur indique un objet à déplacer, puis se rend compte qu'il s'est trompé, il lui suffit alors, pour corriger son erreur, d'indiquer un autre objet tout en prononçant le mot "ça". L'ancien objet sera alors remplacé par le nouveau.
- **Souplesse de la syntaxe d'interaction** Grâce à l'utilisation des coréférences l'utilisateur n'est plus contraint de fournir les données aux commandes qu'il désire exécuter dans un ordre bien précis. En faisant accompagner ces données des références aux arguments correspondants, il pourra alors les produire dans l'ordre qui lui convient le mieux.
Comme vu précédemment, si l'utilisateur désire indiquer d'abord la nouvelle position de l'objet à déplacer, puis l'objet lui-même, il lui faudra alors simplement prononcer le mot "ici" lors du premier geste de pointage et le mot "ça" lors du second.
- **Richesse du feed-back de contrôle** L'entrée des données selon des séquences différentes permet dans certains cas de disposer de différents types de feed-back de contrôle. Par exemple, pour la commande de tracé de cercles, si l'utilisateur décide d'indiquer d'abord le centre du cercle puis son rayon, il pourra dans ce cas disposer d'un contrôle continu sur le rayon. Par contre, s'il indique d'abord le rayon puis le centre, il pourra disposer dans ce cas d'un contrôle continu sur le centre du cercle. Dans le cas de commandes à plus de deux arguments, le contrôle continu peut être obtenu sur le dernier argument indiqué.

Évaluation Au cours de son stage de DEA, D. Fass [15] a réalisé une étude relative au domaine des sciences cognitives en utilisant LIMSI-Draw comme plate-forme multimodale. Cette étude entrait dans le cadre de la collaboration entre les deux groupes Cognition Humaine et Communication Non Verbale du LIMSI. Elle avait pour but principal d'étudier les stratégies cognitives mises en oeuvre par des sujets dans la réalisation de

tâches faisant appel à l'imagerie mentale, dans un contexte multimodal réel (avec les contraintes induites par la technologie actuelle). Nous avons profité de cette occasion pour étudier certains aspects liés plus spécifiquement à l'utilisation d'une interface multimodale. Dans la suite nous reprendrons uniquement les résultats liés à ce deuxième volet de l'étude.

Description de l'expérience Les sujets devaient réaliser quatre tâches différentes de complexité croissante. La première tâche consistait à reproduire sur l'écran de LIMSI-Draw une figure que le sujet avait sous les yeux et qui comportait 9 formes géométriques simples utilisant au maximum 6 couleurs différentes.

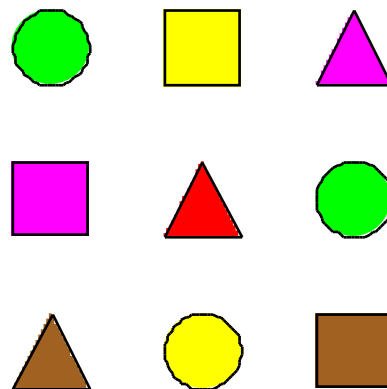


Figure 4 utilisée dans la première tâche.

Dans la seconde tâche, il était demandé aux sujets de mémoriser une figure du même type que la précédente, et de la reproduire ensuite en utilisant LIMSI-Draw.

La troisième tâche consistait à fournir aux sujets une description verbale d'une figure. Ceux-ci devaient alors se construire une image mentale de cette figure, la mémoriser puis la reproduire. Pour aider les sujets, la description de la figure suivait un ordre linéaire :

«En haut à gauche, il y a un rectangle vert.
A droite du rectangle vert, il y a un cercle rouge.
A droite du cercle rouge, il y a un triangle violet.

Sous le rectangle vert, il y a un triangle jaune.
A droite du triangle jaune, il y a un rectangle marron.
A droite du rectangle marron, il y a un cercle blanc.

Sous le triangle jaune, il y a un rectangle blanc.
A droite du rectangle blanc, il y a un triangle rouge.
A droite du triangle rouge, il y a un cercle vert.

La dernière tâche enfin était du même type que la troisième sauf que l'ordre de description était cette fois-ci aléatoire.

L'expérience a été menée avec un ensemble de 16 sujets parmi lesquels figuraient des informaticiens et des non informaticiens (médecin, journaliste, architecte...).

Quelques paramètres étudiés La multimodalité offerte par LIMSI-Draw a permis de mettre à disposition

des utilisateurs différents scénarios d'interaction possibles pour la réalisation de leurs tâches. Ceci nous a permis d'analyser les paramètres décrits ci-dessous.

Relation Média <---> Type information Il s'agit de voir s'il existe une dépendance entre le type de l'information communiquée par l'utilisateur et le média utilisé. On remarque d'après la figure 5 [15] que lorsqu'il s'agit d'exprimer le nom de la commande à exécuter (par exemple "rectangle"), les utilisateurs préfèrent utiliser la parole (95%).

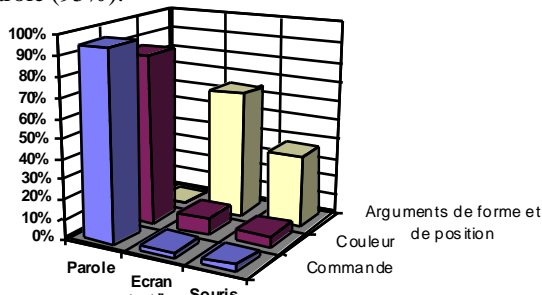


Figure 5. Relation Média <---> Type d'Information

La même remarque est valable pour les noms de couleur avec un pourcentage légèrement inférieur (86%). Enfin concernant les arguments de forme (sommets d'un triangle, centre d'un cercle etc.) et de position, on constate une préférence de la part des utilisateurs pour l'écran tactile (64%) par rapport à la souris (36%), ce qui est assez inattendu vu que la majorité des sujets étaient déjà habitués à l'utilisation de la souris. En revanche on ne peut pas faire de comparaison par rapport à la parole, vu que la possibilité de décrire une position spatiale de manière verbale n'était pas offerte dans LIMSI-Draw. Une analyse plus détaillée montre qu'il y a rarement alternance dans l'utilisation de l'écran tactile ou de la souris. Les utilisateurs choisissent en général soit l'écran tactile, soit la souris, puis n'utilisent plus que le média qu'ils ont choisi pour les opérations de pointage.

Combinaison des médias La figure 6 montre de quelle manière les médias ont été utilisés par les sujets pour réaliser les tâches demandées.

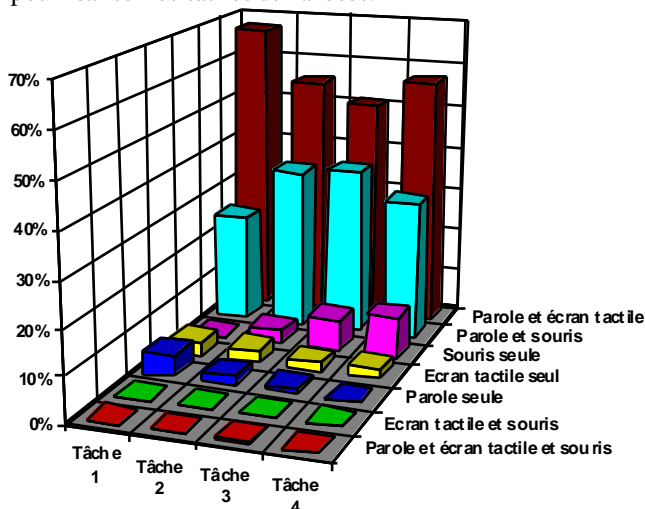


Figure 6 Combinaison des médias.

On remarque tout d'abord que l'utilisation des médias de manière isolée est très faible, la souris ayant le plus haut pourcentage (4,75%).

La combinaison (parole, écran tactile) possède le plus grand pourcentage (57,5%) contre 33,125% pour la combinaison (parole, souris). La combinaison (écran tactile, souris) est inexistante, comme on pouvait facilement s'y attendre, alors que la combinaison (parole, écran tactile, souris) a été utilisée une seule fois par un sujet qui s'est embrouillé un instant.

Séquentialité / Simultanéité (geste, parole)

Il est intéressant de voir de quelle manière les différents sujets ont combiné le geste à la parole.

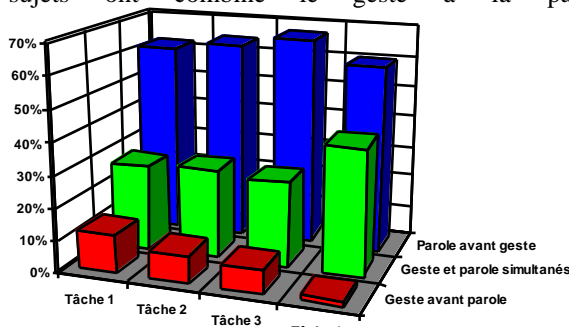


Figure 7 : Séquentialité / Simultanéité (geste, parole)

D'après la figure 7, on peut voir que dans la majorité des interactions, la parole précède le geste (62,25%). Un pourcentage important apparaît pour la combinaison simultanée du geste et de la parole (30,25%). Ce pourcentage aurait très probablement été plus important si le système de reconnaissance vocale était plus performant. En effet, il a été souvent remarqué que les sujets prononçaient le nom d'une commande, vérifiaient sur l'écran que le résultat de la reconnaissance était correct, puis complétaient leur commande en faisant les gestes de désignation nécessaires. Ce manque de confiance par rapport au système de reconnaissance vocale semble donc expliquer le pourcentage élevé des interactions dans lesquelles la parole précède le geste.

Référence et déictique dans IMAR [6, 7]



Figure 8 : Exemple de plan d'aménagement dans le projet IMAR [7]

Dans le projet IMAR (Interface Multimodale d'Applications Robotisées) nous avons proposé une méthode de résolution des références intermodales pour une interface multimodale utilisant les modalités parole et geste de désignation.

La résolution des références intermodales est découpée en deux parties. Une première partie est effectuée dans un processus de compréhension qui a pour rôle de produire une représentation sémantique d'un énoncé. Cette partie, générique à toute application, fournit les référents possibles pour chaque référence [6]. En ce qui concerne les références intermodales, elles sont calculées suivant des critères d'intégration faisant suite à ceux identifiés lors des journées IHM'92 [19] :

- le critère de proximité temporelle qui permet de restreindre la recherche d'un référent dans une fenêtre temporelle autour du moment où ont été dits les mots faisant appel à une référence intermodale.
- le critère d'incompatibilité des modalités décrit les modalités qui ne permettent pas à une certaine modalité d'avoir un référent dans une autre modalité dites incompatibles. Ceci restreint l'espace de recherche des référents à ceux provenant des modalités compatibles.
- le critère de **complémentarité sémantique** : nous utilisons ici le terme sémantique au lieu de logique afin de montrer que ce sont des attributs sémantiques qui servent de contraintes lors de l'attribution des référents aux références déictiques. Le terme logique est dans notre étude trop général et peut porter à confusion.
- le critère de **complémentarité par attributs**, il se distingue du critère par complémentarité sémantique par le fait que les contraintes imposées par ce critère portent sur les attributs sémantiques de la chose désignée et non sur la chose dans son ensemble. Ce critère sera validé si l'utilisateur dit « mets la table de cette couleur » en désignant une chaise. Dans cet exemple, le référent du syntagme « cette couleur » porte sur la couleur de la chaise et non sur la chaise.
- le **critère d'ordonnement** des références; celui-ci est validé si l'ordre temporel des syntagmes contenant des déictiques et l'ordre des objets désignés correspondant à ces déictiques est le même. Par exemple, si un utilisateur dit « pose ce vase sur cette table » en désignant d'abord un vase puis une table, alors ce critère est validé. Ce critère ne serait pas validé si l'ordre des désignations avait été inversé (désignation de la table suivi de celle du vase).

- le **critère de complétude** des événements utilisateur; ce critère est validé lorsque toutes les références intermodales ont été résolues et lorsque tous les événements utilisateur ont pu être liés dans une même structure. Par exemple, ce critère n'est pas validé s'il reste une désignation gestuelle qui n'a pu faire l'objet d'une référence orale.

La seconde partie est faite dans un processus d'interprétation contextuelle qui a pour objectif de trouver le meilleur référent, suivant le contexte d'interaction parmi ceux proposés par le module de compréhension. Dans le contexte applicatif de IMAR, le mot 'ce', par exemple, dans le syntagme 'ce frigo' peut faire référence à trois sources de référents (gestes de désignation, objets visibles de la scène, historique des objets déjà référencés) [7]. C'est alors le processus d'interprétation, après identification contextuelle de l'action voulue par l'utilisateur qui choisira le référent (parmi ceux proposés par la compréhension) le plus adapté à l'action qu'il veut déclencher suivant les paramètres de cette dernière.

L'étude des modes d'usage dans cette interface entre un CU et un CL n'a pas encore été conduite. Toutefois il s'avère que l'usage conjointe du geste de désignation et de l'énoncé oral n'est opératoire qu'après une phase d'apprentissage. Ce module de résolution des références intermodales supporte les différents types [19] de multimodalité à savoir, synergique, alterné, exclusif et concurrent.

CONCLUSION

Cet étude de cas sur l'aménagement d'une cuisine a permis d'aborder toutes les étapes de conception d'un système interactif. En raison du caractère multidisciplinaire des participants, des points de vue différents en fonction de leur thématique de recherche ont fait l'objet de discussions très fructueuses. Parmi ces points, notons le rôle des contraintes qui interviennent dans la résolution du problème de conception, qui influe la présentation de l'interface, par exemple. Quant au modèle de tâche, il a fait l'objet de nombreuses discussions : le faut-il séquentiel, opportuniste, hiérarchique ? Il en est de même pour l'analyse et le traitement de la référence aux déictiques.

La généralisation de ces résultats au domaine de la conception d'IHM implique tout de même que la spécificité du cas traité (ici une tâche de conception) soit vue comme la projection de questions génériques sur une application. Affaire à suivre à IHM'97 ?

BIBLIOGRAPHIE

- 1 M. Arnold et F. Chopin, Utilisation d'un logiciel d'image de synthèse pour l'aménagement d'une cuisine. Modélisation sémiotique de l'interaction multimodale, actes de la conférence IHM'96, Cépaduès-Éditions, 1996, 2 pages.

- 2 M. Arnold, Semiotic analysis of graphical and linguistic data in architectural design activity.- Semiotica, 111 - 3/4 (1996), pp. 245-67.
- 3 F. Bastide, Iconographie des textes scientifiques. Principes d'analyse.-Culture technique, 14 (1985), 133-51.
- 4 N. Bellalem N., L. Romary, Langue et geste pour le dialogue homme- machine finalisé.- Le communicationnel pour concevoir, ed. par J. Caelen et Kh.Zreik, Paris, Europa Productions, 1995, p. 185-201.
- 5 Y. Bellik, Interfaces multimodales : concepts, modèles et architectures, Thèse de Doctorat en Informatique de l'Université de Paris-XI, Notes et Documents LIMSI, N° 95-23.
- 6 E. Brison, N. Vigouroux, Interprétation des événements multimodaux, dans Sixièmes Journées sur l'Ingénierie des Interfaces Homme-Machine, Lille, 8-9 Décembre 1994.
- 7 E. Brison, Stratégies de compréhension dans l'interaction multimodale, Thèse de Toulouse III, Janvier 1997, à paraître.
- 8 J. Caelen, Vers une logique dialogique. Séminaire international de pragmatique, Jérusalem, 1995.
9. N. Carbonell, C. Valot, C. Mignot, P. Dauchy, Etude empirique : usage du geste et de la parole en situation de communication homme-machine, Ergo'IA 94, Actes du 4ème colloque Ergonomie et informatique avancée, pp.128-39.
- 10 T.L. Dean, M.P. Wellman, Planning and Control. Morgan Kaufman Publishers, 1990.
11. C. Faure et K. Chabani, Aménagement des plans de cuisine : interaction au stylo contrôlée par les connaissances, Pré-actes électroniques de la conférence IHM'96, 1996, 2 pages.
- 12 C.Faure et K. Chabani, Aménagement de plans de cuisine : interaction au stylo contrôlée par les connaissances, Rapport Interne ENST 96C005 "Interactions homme-machine : trois études". 1996.
- 13 C. Faure, Les images dans l'interaction homme-machine, Rapport Interne ENST 96C005 "Interactions homme-machine : trois études". 1996.
- 14 C. Faure, M. Arnold, Interaction linguistique pour la conception cooperative de scènes graphiques in Le Communicationnel pour Concevoir, J. Caelen et K. Zreik (eds), pp. 223-242, 1995.
- 15 D. Fass, Stratégies cognitives mises en œuvre dans l'utilisation d'une station multimodale, Mémoire de DEA, Limsi-CNRS, 1994.
- 16 J.M. Floch, Petites mythologies de l'œil et de l'esprit. Pour une sémiotique plastique. - Amsterdam, Hadès-Benjamins, 1985.
- 17 P. Girard et L. Guittet, Aménagement d'une cuisine : de la CAO mécanique à la CAO pour "non utilisateur", pré-actes électroniques de la conférence IHM'96, 1996, 2 pages.
- 18 A.J. Greimas, Du sens II. Essais sémiotiques.- Paris, Le Seuil, 1983.
- 19 IHM'92, Compte rendu des ateliers, Interface Multimodales et Architecture Logicielle, pp. 9-44.
- 20 C. Mignot, Usage de la parole et du geste dans les interfaces multimodales - Etude expérimentale et modélisation", Thèse de doctorat de l'Université Henri Poincaré - Nancy 1.
- 21 L. Nigay, Conception assistée par ordinateur : aménagement d'une cuisine, Actes de la conférence IHM'96, Cépaduès-Editions, 1996, 2 pages.
- 22 J. Piaget, Development and Learning. Piaget Rediscovered, Ripple R.E. and Roccastle V.N. ed., School of Education, Cornell University, Ithaca, New York, 1964.
- 23 P. Pu and D. Lalanne, Human and machine collaboration in creative design, ECAI'96, Budapest.
- 24 S. Robbe, N. Carbonell, P. Dauchy, How do users manipulate graphical icons? An empirical study, Gesture workshop'96, York, Angleterre.
- 25 E. Visser, Organisation of Design Activities : Opportunistic with hierarchical episodes, in Interacting Computer, Vol. 6, N° 3, pp. 239-74, 1994.