



Eléments de convivialité dans un dialogue de commande multimodal

Laurent Romary, Bertrand Gaiffe, Jean-Marie Pierrel

► **To cite this version:**

Laurent Romary, Bertrand Gaiffe, Jean-Marie Pierrel. Eléments de convivialité dans un dialogue de commande multimodal. 1991. <inria-00537742>

HAL Id: inria-00537742

<https://hal.inria.fr/inria-00537742>

Submitted on 19 Nov 2010

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.

Éléments de convivialité dans un dialogue de commande multimodal¹

Laurent Romary
Bertrand Gaiffe
Jean-Marie Pierrel
CRIN-CNRS/INRIA-Lorraine, B.P. 239, 54506 Vandœuvre Lès Nancy
{romary|gaiffe|jmp}@loria.fr

Thème : *Interaction multimodale*

Résumé : Dans cet article, nous présentons différents points que nous avons développé afin de mettre en œuvre un système de dialogue de commande multimodal au dessus d'un logiciel multimédia existant. Jusqu'à maintenant beaucoup d'attention s'est porté sur l'étude des phénomènes référentiels (Kleiber 81,90, Gaiffe 91) parce qu'une bonne compréhension de ceux-ci donne une meilleure continuité au dialogue et donc pour l'utilisateur plus de naturel. A l'opposé (mais nous verrons que ceci est complémentaire) nous abordons ici le problème du traitement des informations prédictives contenues dans un énoncé pour montrer comment leur gestion adéquate permet à la fois de faciliter la compréhension et de guider l'utilisateur profane ou simplement étourdi dans sa tâche. Notre objectif étant à terme d'intégrer ces phénomènes aux phénomènes référentiels dans un souci d'amélioration de la convivialité des systèmes de dialogue de commande multimodaux.

1. Introduction

1.1 Structure des énoncés

Dans le cadre de la mise en œuvre de nouvelles interfaces destinées à assurer un échange optimum entre un usager et un système informatique, de nombreuses options peuvent être choisies, de l'amélioration des systèmes de fenêtrage à l'étude de nouveaux modes d'interaction (tablette graphique, écran tactile, gant de désignation). La perspective que nous avons adoptée correspond à chercher à prendre en compte au maximum les dispositions naturelles d'un utilisateur à utiliser le langage et éventuellement une désignation simple pour communiquer. Dans cette perspective, il est possible de décrire la structure d'un énoncé élémentaire d'un utilisateur comme la juxtaposition de deux informations : une proposition (au sens large par opposition au sens technique que prend ce terme en logique) et un acte de langage. Dans le cadre d'un dialogue de commande - plus réduit à ce titre qu'un dialogue d'interrogation de base de donnée par exemple - l'acte de langage se limite aux modalités d'ordre, d'affirmation (apport d'information) et éventuellement de question. On est loin des descriptions plus exhaustives faites par exemple par J.Searle (1972).

Notre intérêt va plutôt se porter sur la proposition contenue dans un énoncé de commande pour laquelle on peut mettre en évidence une partie prédictive contenant l'action que la tâche associée au dialogue doit effectuer et une partie référentielle permettant d'établir la liste des objets concernés par le prédicat.

¹ Ce travail a été mené dans le cadre du projet Esprit Multiworks.

L'objectif de cet article est de montrer qu'il est possible à partir des parties référentielle et surtout prédicative d'un énoncé de mettre en œuvre un certain nombre de méthodes visant à assurer une meilleure convivialité à l'interface fournie à l'utilisateur, ces méthodes nécessitant l'établissement de structures de représentation originales prenant en compte à la fois des aspects temporels et référentiels.

1.2 Présentation générale du domaine.

Le cadre général de ce travail est la réalisation d'un gestionnaire de dialogue formant une interface supplémentaire pour un logiciel existant. Le logiciel que nous avons choisi : *Multicard* (un éditeur de document multimédia développé par Bull), présente différentes caractéristiques qui nous permettent d'envisager la mise en œuvre d'un dialogue relativement élaboré avec celui-ci. Tout d'abord, la tâche associée au logiciel est suffisamment simple et consiste essentiellement à créer, détruire et modifier des objets qui formeront la base du futur document multimédia. Dans la version actuelle du logiciel, ces commandes sont accessibles à l'utilisateur par l'intermédiaire d'une interface de type Macintosh (menus) que le dialogue envisagé doit rendre plus convivial. Mais par ailleurs, les objets apparaissent en nombre important, et surtout, sont de types très différents, ce qui impose à l'utilisateur une charge cognitive importante pour gérer son environnement. Ainsi, la mise en œuvre s'en trouve complexifiée puisqu'il s'agit de procurer à l'utilisateur, dans le cadre d'un dialogue multimodal, tous les atouts nécessaires à une bonne spécification des objets sur lesquels il désire travailler.

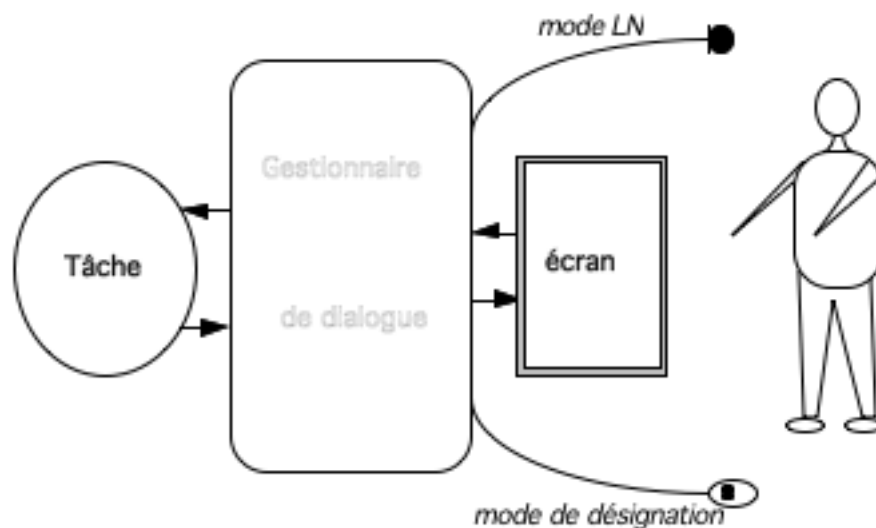


Figure 1.1 : Articulation du dialogue avec la tâche et les différents modes.

A ce stade nous pouvons donner une idée générale de la manière avec laquelle un tel gestionnaire de dialogue doit s'articuler avec la tâche et les différents modes (figure 1.1). Dans cet environnement, l'utilisateur et le système peuvent échanger des informations de différentes manières. Pour transmettre de l'information au système, l'utilisateur peut soit utiliser un mode Langage Naturel, implémenté par exemple sous la forme d'une entrée parole ou d'un clavier (du fait des difficultés liées actuellement à la première solution), soit un mode de Désignation, mis en œuvre actuellement par l'intermédiaire d'une souris, mais pour lequel un véritable gant de désignation serait une solution plus naturelle (certains points envisagés au niveau de la référence se conçoivent d'ailleurs mieux avec une telle entrée).

1.3. Eléments de continuité du dialogue.

Il est important de situer les énoncés d'un dialogue les uns par rapport aux autres à plus d'un titre. Certains problèmes de référence à des objets (anaphores pare exemple) ne peuvent se résoudre que dans cette continuité. Nous verrons dans la suite en quoi les références à des actions entrent aussi dans ce cadre de continuité via l'aspect gestion de plan.

Les éléments classique de continuité dans le dialogue relève de différentes approches :

- rhétorique lorsqu'il s'agit d'argumentation (Moeschler 1989), qui ne nous intéresse que peu dans le cadre de dialogue de commande.
- cohérence dans la référence aux objets, les énoncés étant liés quand ils parlent des mêmes objets (c'est un point essentiel pour la résolution des anaphores).
- cohérence dans la référence aux actions : les actions successives induites par les énoncés peuvent refléter un plan de l'utilisateur. Détecter ce plan aide évidemment à gérer le dialogue et en particulier à résoudre des anaphores du type "refait le" ou "recommence". La suite de cet article montrera en quoi cela permet également de guider l'utilisateur dans sa tâche.

Evidemment, les deux derniers points influent l'un sur l'autre. Dans un plan de l'utilisateur, un certain nombre d'objets ou de types d'objets sont nécessairement récurrent. Enfin, dernière remarque, la cohérence dans la référence aux objets en lien avec les actions suppose une représentation de la vie des objets qui reflète leurs différents états. Nous évoquerons ces deux aspects de gestion des propriétés des objets de la tâche et de gestion de la continuité des actions (plan) dans les deux sections suivantes.

2. Gestion des informations temporelles.

2.1. Mise à jour des propriétés des objets de la tâche.

Comme nous l'avons vu, une commande élémentaire au système est composée d'un opérateur et de n objets sur lesquels doit s'appliquer celui-ci (au sens large, puisque nous incluons ici les objets qui doivent être créés par la commande courante). Le problème de la référence, traité dans (Gaiffe, 1991), ne sera pas repris ici. Il est cependant important de retenir qu'une convivialité maximum est obtenue dès lors que l'on donne à l'utilisateur de multiples possibilités d'effectuer cette opération, avec en particulier la possibilité de réaliser des anaphores (propres au mode LN) ou des coréférences multimodales. La compréhension de tels expressions ne peut être faite de manière convenable qu'à la condition de mettre en œuvre un mécanisme de focus généralisé par rapport à celui introduit par Sidner (81,83, Grosz 86) et schématisé figure 2.1. Si nous supposons ici résolus les problèmes de calcul de référence, nous disposons donc à ce stade d'un certain nombre de propriétés pour chacun de ces objets, propriétés qui vont être modifiées par l'application de la commande au niveau de la tâche. Il est important à ce stade de considérer les deux aspects de cette mise à jour en mettant en évidence les deux lieux où de l'information relative aux objets est enregistrée. En premier lieu, la tâche (ici Multicard) a ses propres structures de donnée et gère, pour chaque action, les modifications adéquates. Ces modifications peuvent être visibles à

l'utilisateur (elles se traduisent alors par une modification de l'écran) ou invisibles (réorganisation éventuelle de la mémoire, ajout d'attributs liés au système etc...). Le dialogue ne doit s'occuper de ces modifications que dans la mesure où elles sont pertinentes pour l'utilisateur (i.e. l'utilisateur risque de s'en servir dans ses énoncés ultérieurs). De fait, le gestionnaire de dialogue doit posséder sa propre représentation de ces informations, représentation essentiellement symbolique, qui doit à la fois traduire un certain état graphique présent devant l'utilisateur et permettre une liaison avec les instructions en provenance du Langage Naturel².

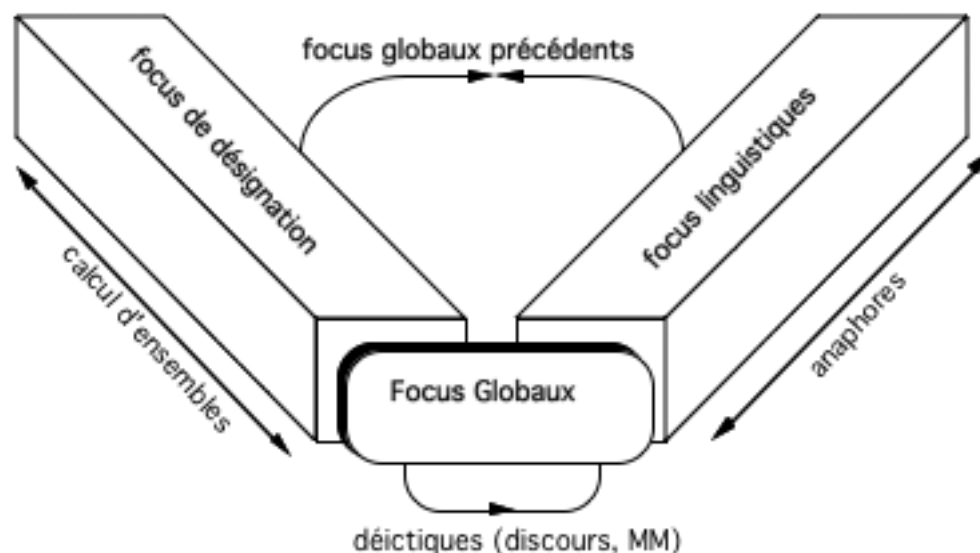


Figure 2.1 : Structures de focus associées à une référence multimodale.

Deux stratégies peuvent être alors envisagées : soit interroger régulièrement la tâche pour obtenir des renseignements sur les états respectifs des différents objets présents, soit tenir à jour - par une modélisation adéquate des actions - les propriétés des objets en supposant qu'un parallèle correct est établi entre les opérations réelles de la tâche et les "modèles" de celles-ci seuls connues du dialogue. La première solution, bien que plus précise, nécessite la mise en œuvre de procédures de filtrage et de symbolisation d'informations arbitraires (puisque liées à un logiciel préexistant au dialogue), mais bien plus, elle impose une dépendance extrême du dialogue vis à vis de la tâche. L'autre démarche nous semble maintenant plus profitable car elle permet d'une part de disposer d'une information proche de celle manipulée par l'utilisateur dans ses énoncés et d'autre part, elle impose de disposer d'une représentation des opérateurs de la tâche qui, comme nous le verrons dans la section suivante, sont autant de briques élémentaires pour le calcul de plan d'aide à la réalisation des objectifs de l'utilisateur.

2.2. Reconnaissance et proposition de plans.

Dans un certain nombre de travaux sur le dialogue (Agre, 1989) la notion de plan est introduite pour refléter la stratégie adoptée par le locuteur pour construire son discours. Dans le cadre d'un dialogue de commande tel que celui que nous considérons ici, cette notion nous paraît bien plus importante quand elle est reliée à la tâche elle-même parce qu'il est alors possible de mieux suivre les énoncés successifs de

² On pourra parler alors de véritable représentation de connaissances multimodales.

l'utilisateur et même parfois de lui proposer une aide dans le cadre de sa tâche. Cette gestion des plans est rendue nécessaire du fait que contrairement au dialogues gérés par Grosz (1986), l'arbre des sous-tâches n'est pas préexistant au dialogue mais construit au fur et mesure par l'utilisateur.

- *reconnaissance de plan*

L'objectif est ici de situer l'action courante de l'utilisateur dans une perspective un peu plus large qui facilite la compréhension des énoncés à venir. Ceci d'une part en reliant les actions qui peuvent se correspondre (par exemple : vérification des préconditions d'une nouvelles action dans le cadre restreint de la précédente) et d'autre part de procurer des valeurs par défauts déduites du contexte dans l'hypothèse où effectivement l'utilisateur suit un plan cohérent. En fait de plan, nous considérons dans un premier temps des scripts d'action réduits tel que celui présenté figure 2.1, où deux actions sont considérées, une action de création de groupe (i.e. un ensemble de nœuds dans Multicard) suivie d'une création de nœud, pour laquelle il est possible d'ajouter une propriété par défaut, à savoir l'appartenance du nœud au groupe qui vient juste d'être créé.

créer_groupe(x0)	créer_nœud(y0)
	y0 ∈ x0

Figure 2.2 : script élémentaire de l'utilisateur.

- *proposition de plan*

Le calcul de plan pour l'utilisateur se place dans une perspective complémentaire d'aide à la mise en œuvre d'un document Multicard. En effet, un utilisateur de Multicard ne connaît pas nécessairement toutes les préconditions associées à chaque action et même dans le cas contraire, il ne désire pas nécessairement énoncer tous les détails d'une opération menant à la création de tel ou tel objet. La solution de refuser systématiquement tous les ordres qui ne sont pas strictement applicables dans un contexte s_0 fixé n'est pas nécessairement la meilleure pour assurer une bonne convivialité à l'ensemble. Aussi avons nous choisi de permettre à l'utilisateur (jusqu'à un certain point) de se limiter à l'expression de jalons élémentaires dans la mise en œuvre d'un document. Le principe est le suivant : si s_0 est l'état courant (les objets et leurs propriétés) et si P est une commande de l'utilisateur, si $P(s_0)$ est applicable, alors on exécute simplement la commande, sinon, on propose à l'utilisateur une suite P_1, P_2, \dots, P_n d'actions élémentaires permettant d'atteindre les préconditions de P, à savoir tels que $P P_n \dots P_1(s_0)$ soit réalisable. L'utilisateur valide alors cette séquence si elle correspond à son intention initiale. A titre d'exemple supposons simplement que l'utilisateur exprime l'action de supprimer un nœud qui s'avère contenir un lien avec un autre objet du document courant. Le système propose alors de supprimer au préalable ce lien afin d'atteindre l'une des préconditions de suppression d'un nœud qui est que celui ci soit vide de tout lien.

Il faut bien garder à l'esprit que les deux niveaux de plan que nous présentons ici présentent une granularité de description des actions assez différente. Dans le cas de la

reconnaissance de plan, on se contente de relier des actions importantes susceptibles d'interagir. Dans le second cas chacune des actions élémentaires est déterminée et est proposée à l'utilisateur.

5. Conclusion.

Les différents éléments présentés ci-dessus ont essentiellement concerné le traitement de l'information prédicative d'une commande d'un utilisateur. Cependant, il est indispensable de ne pas séparer ce niveau d'information de celui de la référence qui participent conjointement, comme le fait justement remarquer J.Hobbs (1979), à la structuration et à la continuité du discours. Nous avons vu dans cet article que la représentation de certaines informations temporelles liées au traitement des actions était un point de passage obligé lors de la mise en œuvre d'un système de dialogue de commande. Notre objectif est maintenant d'intégrer ces informations dans une représentation communes des objets référencés dans le discours et des événements qui leur sont associés. Cette étape, qui doit prendre en compte nos travaux en cours sur la représentation du temps (romary 91), permettra de réaliser une intégration similaire à celle réalisée dans le cadre de la DRT par B.Partee (1984) par exemple. Enfin, signalons que ces différents axes de recherche font l'objet d'une attention particulière dans le cadre de la mise en œuvre d'une nouvelle maquette pour le projet Multiworks.

6. Références.

- Agre, P. and Chapman, D. (1989), "What are plans for ? In P.Maes (ed.) *Designing Autonomous Agents : Theory and Practice from Biology to Engineering*. Cambridge MA: MIT Press, 1991.
- Carbonell N. et Pierrel J.M., 1989, "Vers un dialogue naturel homme-machine : apport des études sur les interfaces orales en langue naturelle", *Actes du colloque sur l'ingénierie des interfaces homme-machine*, Cargèse.
- Gaiffé B., Romary L. & Pierrel J.M., 1991, "References in a Multimodal Dialogue : towards a unified processing", *Proc. Eurospeech 91*, Gènes.
- Grosz B. and Sidner C., 1986, "Attention, Intentions and the structure of discourse", *Computational Linguistics*, **12**, pp.175-204.
- Hobbs J.R., 1979, "Coherence and coreference", *Cognitive Science*, **3**(1), pp.67-90.
- Kleiber G., 1981, *Problèmes de référence : descriptions définies et noms propres*, Klincksieck, Paris.
- Kleiber G., 1990, "Sur l'anaphore associative : article défini et adjectif démonstratif.", *Rivista di linguistica*, **2**, n°1.
- Moeschler J., 1989, *Modélisation du dialogue : représentation de l'inférence argumentative*, Hermès, Paris.
- Partee B., 1984, "Nominal and temporal anaphora", *Linguistics and Philosophy*, **7**, pp.243-286.
- Pierrel J.M., 1990, "Vers une meilleure intégration de la parole dans des systèmes de communication homme-machine", *Traitement du signal*.
- Romary L., 1991, "Integration of spatial and temporal information produced by a natural language discourse", *in proc. Kmet 91*, Sophia-Antipolis, 22-24 avril 1991.
- Searl J.R., 1972, *Les actes de langage*, Hermann, Paris.
- Sidner C., 1981, "Focusing for interpretation of pronouns", *American Journal of Computational Linguistics*, **4**, pp.217-231.
- Sidner C., 1983, "Focusing and discourse", *Discourse Processes*, **6**, pp.105-142.