

Une Logique pour le Dialogue Coopératif Homme-Machine

Jean Caelen

Laboratoire CLIPS-IMAG
Domaine Universitaire,
BP 53, 38041 Grenoble Cedex 9

RÉSUMÉ

L'objectif du dialogue homme-machine (DHM) est de donner à la machine des capacités accrues de communication et d'adaptation au locuteur. Dans le cadre du dialogue coopératif finalisé, la réalisation de la tâche structure le dialogue. C'est à travers le dialogue que s'établit *quoi* faire et aussi *comment* le faire. Notre idée est de fonder le modèle du dialogue sur la notion d'apprentissage des savoirs et des savoir-faire où la notion d'action est primordiale. Cet article présente les premières considérations pour la gestion de ce type d'échanges dialogiques en définissant une logique pour le dialogue coopératif homme-machine.

MOTS CLÉS : logique dialogique, dialogue coopératif, gestion de buts, interaction multimodale.

1. Introduction

On peut distinguer deux grandes voies de développement dans les interfaces personne-système [7, 11]. D'une part, les interfaces à manipulation directe qui adoptent une métaphore où la machine est un outil virtuel que l'utilisateur emploie pour réaliser sa tâche. Et d'autre part, les interfaces à composante langagière avec lesquelles l'utilisateur dialogue en parlant. Dans le premier cas, l'utilisateur a la sensation d'être le responsable direct de la manipulation des objets et le système doit devenir alors le plus possible transparent pour donner à l'utilisateur toute la liberté possible dont il a besoin pour agir. Dans le deuxième cas, l'utilisateur devient un locuteur qui parle de sa tâche et la délègue à la machine qui adopte alors le rôle d'un assistant. Bien sûr cet assistant doit être doté de capacités de compréhension de la langue naturelle mais aussi doit connaître les règles du dialogue ou de la conversation. Notre travail tente d'intégrer ces deux types d'interfaces en prenant comme métaphore principale le dialogue coopératif orienté par la tâche. Pour cela, nous développons ci-dessous une logique de dialogue en y intégrant les actes de langage. Les actes de langage peuvent s'assimiler à des *faire-faire* (notés ff) tandis que les actions gestuelles sont des *faire* (notés f). Une logique peut donc se construire sur une théorie unifiée de l'action.

1.1 La Langue, le Dialogue et l'Action

« Le langage se construit par l'action » : c'est un des résultats principaux de Piaget [8]. Pour lui, l'enfant construit son langage comme résultat de l'assimilation des actions sur le monde. Appeler cette personne « maman » ne fonctionne que si celle-ci accourt au cri de « maman ».

Réciproquement « Le langage construit l'action ». C'est la thèse principale de l'école anglo-saxonne en philosophie du langage [1, 10]. Dans cette théorie, parler c'est agir, produire des actes ; communiquer, c'est agir sur l'interlocuteur. Cette théorie permet de considérer le langage comme une forme d'action et, par généralisation, de considérer le dialogue comme une séquence d'actions planifiées ayant pour objectif un *but visé sous-tendu par une intention*. Cette conception présume qu'il existe un *équilibre rationnel* entre les connaissances, actions et intentions du locuteur. Par exemple, on espère que la série d'actions que l'utilisateur est en train de faire, coïncide avec la réalisation de ses intentions, et qu'en effet, l'utilisateur adopte seulement des intentions qui sont possibles à réaliser.

« Le dialogue est une interaction : il renvoie le langage à l'action et réciproquement ». Le dialogue est une suite coordonnée d'actions (langagières et non-langagières) devant conduire à un but [16]. Ce but doit être à la fois atteint et satisfait pour que le dialogue aboutisse à un succès [15]. Le dialogue avance dans le temps et tend à réduire les écarts entre les interlocuteurs. Ceux-ci agissent sous forme de requêtes, répliques, réponses, mises en question ou même remises en cause (figure 1). Au cours de cette interaction ils modifient leurs connaissances, leurs croyances, acquièrent de nouvelles connaissances tant sur la situation, que sur leur interlocuteur ou sur la langue.

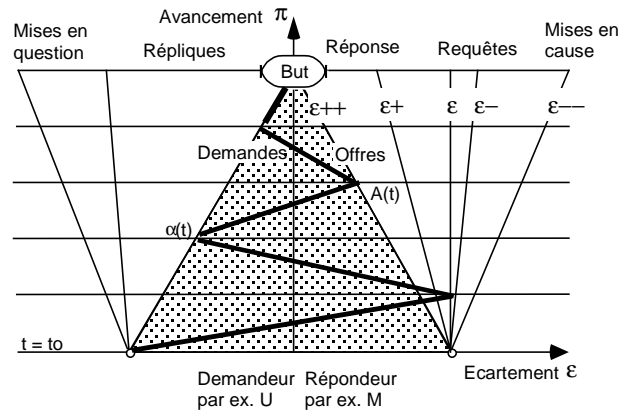


Figure 1. Convergence du dialogue au cours du temps : le modèle projectif de Vernant [16]

1.2 Action et Langue : Que Retenir ?

Notre hypothèse est que pour favoriser au maximum l'interaction entre l'homme et la machine, il faut donner à cette dernière des capacités d'action par la langue mais aussi laisser l'opportunité à l'utilisateur d'agir par manipulation directe. La seule structure qui peut alors lier les deux sortes d'interaction est le dialogue qui est vu comme un double processus de conduite interactive de la tâche et d'acquisition de connaissances langagières utiles à l'action :

- (a) d'une part, la machine doit acquérir les concepts manipulés à travers le langage et qui sont souvent « naturels » (donc implicites) pour l'utilisateur,
- (b) d'autre part elle doit apprendre de nouveaux plans d'action dans le contexte d'usage donné et de manière suffisamment générique pour être réutilisables.

En effet, nous n'avons plus à distinguer, du point de vue de l'utilisateur, une action élémentaire qu'il fait de manière gestuelle et une action complexe qu'il fait par la langue et qui pourrait se ramener à une série d'actions gestuelles. Le niveau de granularité d'une action gestuelle est souvent plus fin que celui d'un acte de langage. Par exemple pour dessiner une maison, il suffira peut-être d'ordonner "dessine une maison" alors qu'un dessin gestuel nécessiterait une série d'actions différentes pour tracer le corps de la maison puis le toit, puis les fenêtres, etc. Ainsi nous serons amenés à distinguer deux niveaux de granularité dans les actions : les instructions élémentaires et les plans construits (ou tâches). L'énoncé "dessine une maison" sera donc associé à un plan d'action décomposable en une série d'instructions élémentaires. De la sorte, la machine pourra faire des actions nouvelles (sorte de macro-instructions) à partir de ce corps d'instructions élémentaires (sorte d'instructions de base dans un langage de programmation) en y associant des nouveaux concepts comme "dessine", "maison", etc. – nous ne voulons pas considérer pour autant le langage naturel comme une sorte de super-langage de programmation car cette approche réduirait le dialogue à une entrée interactive de programmes, il perdrait sa spontanéité et son aspect coopératif (il adopterait la structure des langages impératifs ou des langages déclaratifs).

1.3 Position Théorique du Problème

Notre modèle s'appuie sur la théorie des actes de langage [1, 10] qui, parmi les différents paradigmes de la communication, semble le plus adéquat pour une modélisation du dialogue homme-machine. Cette question a été souvent soulevée nous n'y reviendrons pas ici. Notre modèle fait l'hypothèse que le dialogue est dirigé par les *états mentaux* qui sous-tendent les croyances, les actions et les intentions de l'utilisateur. Ce modèle ne vise pas à donner un comportement de type humain à la machine mais seulement à lui fournir des éléments logiques de choix et des raisons d'agir [2].

L'action immédiate et l'action différée : ce qui distingue essentiellement une action gestuelle est qu'elle est la plupart du temps immédiate (j'appuie sur le bouton électrique, la lumière s'allume) et que ses effets ne sont pas toujours réversibles. Dans le cas d'une action langagière il n'y a plus cette immédiateté car l'action langagière doit être médiatisée par un agent agissant. Les effets sont donc réversibles pendant une fraction de temps où l'agent n'a pas encore agi. Les effets ne sont pas toujours prévisibles car ils dépendent de la décision d'agir et du choix de l'action de l'agent : il peut se tromper, ne pas comprendre, faire une fausse manœuvre, etc. Les facteurs qui distinguent donc l'action gestuelle de l'action langagière sont :

- a) le temps de réaction entre l'émission de l'ordre et l'obtention de son résultat,
- b) la pertinence des effets obtenus. La pertinence n'a de sens que par rapport aux intentions préalables du donneur d'ordre.

Ainsi dans le dialogue on doit contrôler l'évolution du but : entre le moment où il est posé et le moment où il est satisfait, il peut arriver beaucoup d'événements.

Une logique pour le dialogue : la logique que nous visons, tente la modélisation des échanges dialogiques entre l'utilisateur et la machine en utilisant le concept des actes de dialogue. L'idée principale est l'intégration des actes de dialogue dans un cadre logique fondé sur l'action qui permet expliquer le déroulement d'un dialogue homme-machine. Ce travail s'inspire des travaux proposés par [2, 4, 6, 9, 12, 13, 14]. Il contient des éléments d'une logique épistémique –pour la représentation des connaissances–, d'une logique de l'action –pour décrire les actions et les effets produits par ces actions–, et d'une logique dialogique –pour exprimer les engagements, les interruptions et les incompréhensions lors des échanges dialogiques, en fonction du buts du dialogue.

2. Concepts de Base

La *connaissance* est représentée par l'opérateur *s* (savoir) p. e. la formule $Us \varphi$ exprime que l'utilisateur sait la proposition φ (pour faire la distinction entre l'utilisateur et la machine, les deux agents possibles dans notre logique, on utilise les lettres *U* pour l'utilisateur et *M* pour la machine). Pour l'action on introduit la notion de 'événement' : un événement $Uf \alpha$ est la réalisation de l'action α par l'utilisateur (ou $Mf \alpha$ par la machine) et cet événement a la proposition φ comme résultat, en utilisant la notation de la logique de l'action on a la formule suivante $[Uf \alpha] \varphi$ (après avoir terminé l'exécution de α par l'utilisateur, φ est vrai). On représente l'intention par l'opérateur *i*, uniquement si l'utilisateur est capable d'agir intentionnellement, ainsi la formule $Ui \varphi$ exprime l'intention de l'utilisateur de rendre φ vrai. Finalement une action peut être (i) une action de base (instruction élémentaire) ou (ii) une tâche, c'est-à-dire, une séquence d'actions (organisée à travers un plan).

2.1 L'Acte de Dialogue

Un acte de dialogue est une action qui entraîne un changement de la situation par rapport à la tâche (et/ou de la connaissance en machine sur la tâche) et un changement de la situation du dialogue. Ainsi un acte de dialogue est défini par un événement $[Uf \alpha](\varphi_\alpha \wedge \varphi_d)$ où le résultat est l'ensemble des changements de situation par rapport à la tâche φ_α et par rapport à la situation du dialogue φ_d . Les changements par rapport à la tâche sont les effets propres de l'action (φ_α). Et les changements par rapport au dialogue sont les conséquences qui montrent l'avancement du but à chaque tour de parole. Dans notre cas le but est la tâche que l'utilisateur cherche à accomplir. Ainsi, chaque acte de dialogue a comme partie de sa réalisation un effet sur le but (φ_d). Cet effet, qui explicite l'évolution du but, est annoté de la manière suivante :

- ? but posé, action (ou plan) à faire
- + but atteint, action réalisée sans confirmation du demandeur mais vraisemblablement correcte
- ++ but satisfait, action achevée correcte et confirmée par le demandeur
- @ but abandonné

Les actes de dialogue sont exprimés à partir des actions de la manière suivante :

$Uf \alpha, Mf \alpha$	l'utilisateur ou la machine <i>fait</i> une action α
$Uff \alpha, Mff \alpha$	l'utilisateur <i>fait faire</i> une action α à la machine (ou vice versa)
$Ufs \phi, Mfs \phi$	l'utilisateur <i>fait savoir</i> ϕ à la machine (ou vice versa)
$Uffs \phi, Mffs \phi$	l'utilisateur <i>fait faire-savoir (=demande)</i> ϕ à la machine (ou vice versa)

où l'action **fs** est une abréviation de l'action de base **partager** : $Ufs \phi \equiv Uf \text{ partager } \phi$, car faire partager est la conséquence attendue du locuteur sur son allocataire.

3. Définition du Langage L_d

3.1. DEFINITION. Si T est un ensemble dénombrable de symboles propositionnels, Ab un ensemble fini d'actions de base, U le symbole pour dénommer à l'utilisateur et M pour la machine, le langage L_d est défini de la façon suivante :

L_d est le plus petit ensemble de T tel que :

- si $\phi, \psi \in L_d$ alors $\neg\phi, \phi \vee \psi \in L_d$
- si $\alpha \in Ac$, et $\phi \in L_d$ alors $Us \phi, Ms \phi, [Uf \alpha]\phi, [Mf \alpha]\phi, Ui \phi \in L_d$

où Ac est le plus petit ensemble de Ab tel que :

- si $\alpha \in Ab$ alors $\alpha \in Ac$
- si $\phi \in L_d$ alors $\text{vérifier}(\phi) \in Ac$
- si $\alpha \in Ac$ et $\beta \in Ac$ alors $\alpha; \beta \in Ac$

On utilise les abréviations $\phi \wedge \psi$ pour $\neg(\neg\phi \vee \neg\psi)$ et $\phi \supset \psi$ pour $\neg(\phi \wedge \neg\psi)$. On prend *vrai* comme une abréviation d'une formule valide, p. e. $\phi \vee \neg\phi$, et *faux* comme une abréviation de $\neg\text{vrai}$.

3.2. DEFINITION. Sémantique pour le langage L_d . La classe \mathbf{M} de modèles de Kripke contient tous les tuples $M = \langle S, \pi, R_U, R_M, I_U, r_U, r_M \rangle$ tel que

- (i) S est un ensemble de mondes possibles, ou d'états.
- (ii) π est une fonction qui assigne des valeurs de vérité aux symboles propositionnels T dans un monde possible s . ($\pi(s) : T \rightarrow \{1,0\}$ pour tout $s \in S$)
- (iii) R_U est une relation binaire entre les mondes possibles de S , c'est la relation d'accessibilité associée aux connaissances de l'utilisateur. ($R_U \subseteq S \times S$)
- (iv) R_M est une relation binaire entre les mondes possibles de S , c'est la relation d'accessibilité associée aux connaissances de la machine. ($R_M \subseteq S \times S$)
- (v) I_U est une relation binaire entre les mondes possibles de S , c'est la relation d'accessibilité associée aux intentions de l'utilisateur. ($I_U \subseteq S \times S$)
- (vi) r_U est une relation entre un ensemble de mondes possibles causés par la réalisation d'une action α fait par l'utilisateur dans un monde possible s . ($r_U : Ac \times S \rightarrow \wp(S)$)
- (vii) r_M est une relation entre un ensemble de mondes possibles causés par la réalisation d'une action α fait par la machine dans un monde possible s . ($r_M : Ac \times S \rightarrow \wp(S)$)

3.3. DEFINITION. Soit $M = \langle S, \pi, R_U, R_M, I_U, r_U, r_M \rangle$ un modèle de Kripke de la classe \mathbf{M} . Le concept de vérité d'une proposition (\models) dans un monde possible s à partir du modèle M est inductivement défini de la manière suivante :

$M, s \models \phi$	ssi (si et seulement si) $\pi(s)(\phi) = 1$ pour $\phi \in T$
$M, s \models \neg\phi$	ssi $M, s \not\models \phi$
$M, s \models \phi \vee \psi$	ssi $M, s \models \phi$ ou $M, s \models \psi$
$M, s \models [Uf \alpha] \phi$	ssi $\forall s' [s' \in r_U(\alpha, s) \Rightarrow M, s' \models \phi]$
$M, s \models [Mf \alpha] \phi$	ssi $\forall s' [s' \in r_M(\alpha, s) \Rightarrow M, s' \models \phi]$
$M, s \models Us \phi$	ssi $\forall s' [(s, s') \in R_U \Rightarrow M, s' \models \phi]$
$M, s \models Ms \phi$	ssi $\forall s' [(s, s') \in R_M \Rightarrow M, s' \models \phi]$
$M, s \models Ui \phi$	ssi $\forall s' [(s, s') \in I_U \Rightarrow M, s' \models \phi]$

où r_A où $A = \{U, M\}$ est définie par

$$\begin{aligned} r_A(\text{vérifier}(\phi), s) &= \{s\} \text{ si } M, s \models \phi \\ &= \emptyset \text{ sinon} \\ r_A(\alpha; \beta), s &= r_A(\beta, r_A(\alpha, s)) \end{aligned}$$

4. Définition des Axiomes

Les paragraphes suivants présentent les axiomes de notre logique : une première partie présente brièvement les axiomes qui caractérisent la connaissance et, une deuxième partie introduit les concepts associés à la conduite du but dans le cas du dialogue coopératif.

4.1 Caractérisation de la Connaissance

Les axiomes suivants tentent de caractériser la connaissance en machine : sur les tâches (le savoir-faire requis par la réalisation d'une tâche) et, sur les intentions de l'utilisateur (inférées à partir des demandes ou questions posées, indispensables pour la gestion du dialogue). Cependant, ils sont écrits de façon générale, c'est-à-dire du point de vue de l'utilisateur ou de la machine¹. Pour $A = \{U, M\}$:

¹ Pour une présentation détaillée voir le système S5 dans [6].

(A1)	$\mathbb{A}s \varphi \wedge \mathbb{A}s(\varphi \supset \phi) \supset \mathbb{A}s \phi$	axiome K
(A2)	$\mathbb{A}s \varphi \supset \varphi$ on sait seulement des faits vrais	
(A3)	$\mathbb{A}s \varphi \supset \mathbb{A}s \mathbb{A}s \varphi$ on sait qu'on sait	introspection positive
(A4)	$\neg \mathbb{A}s \varphi \supset \mathbb{A}s \neg \mathbb{A}s \varphi$ on sait qu'on ne sait pas	introspection négative

4.2 Le Déroulement du Dialogue

Dans notre modèle, le déroulement du dialogue est réglé par l'intention de l'utilisateur. On suppose qu'à un moment donné de l'échange, un acte de dialogue de l'utilisateur exprime une intention (si ce n'est pas le cas, la machine posera une question le moment venu). Cette intention de l'utilisateur est interprétée comme l'action souhaitée à exécuter (ou le plan d'action à développer) dont l'effet attendu est le *but* à atteindre. Ainsi, la réalisation d'un acte de dialogue fait normalement converger vers le but à atteindre, sauf s'il s'agit d'un acte qui provoque un changement dans l'évolution du but (abandon, mise en attente, etc.). Cette évolution dépend de la situation actuelle et de l'acte même. Examinons cela en détails.

4.2.1 But Posé par l'Utilisateur

Un but est *posé* par l'utilisateur quand il fait soit une demande de réalisation d'une action *Uff*, soit une demande d'information *Uffs*. Ce but est l'état initial du processus. Il est rangé dans une pile et va subir des transformations. A l'état final, le but est *atteint*, si la machine répond *Mfs* ou exécute l'action demandée *Mf*, puis il est *satisfait* (c'est-à-dire sorti de la pile) quand l'utilisateur exprime son avis, de manière explicite ou implicite, sur l'adéquation de l'action ou la réponse obtenue. Un but peut-être aussi *abandonné* en cours de route par l'utilisateur quand il communique explicitement son changement d'avis. Pour le cas où l'utilisateur demande la réalisation d'une action *Uff*, on a les notations suivantes :

- i) But *posé*, l'abréviation $?Mf \alpha$ désigne l'action *Mf* α comme but posé.

$$[Uff \alpha] (Ms Ui [Mf \alpha] \varphi) \equiv [Uff \alpha] (? [Mf \alpha] \varphi)$$

un but posé est l'effet d'une demande et cet effet est exprimé comme l'intention de l'utilisateur (pour que la machine fasse une action) intégrée aux connaissances en machine.

- ii) But *atteint*, l'abréviation $+Mf \alpha$ désigne l'action *Mf* α comme but atteint.

$$\begin{aligned} Ms Ui [Mf \alpha] \varphi \wedge [Mf \alpha] \varphi \\ \equiv ? [Mf \alpha] \varphi \wedge [Mf \alpha] \varphi \\ \equiv + [Mf \alpha] \varphi \end{aligned}$$

un but posé est atteint quand l'action demandée par l'utilisateur est aussi vraie.

- iii) But *satisfait*, l'abréviation $++Mf \alpha$ désigne l'action *Mf* α comme but satisfait.

$$\begin{aligned} Ms Ui [Mf \alpha] \varphi \wedge ([Ufs U-i [Mf \alpha] \varphi] \vee ([Uff \beta] \gamma \wedge \neg \text{relation}(\varphi, \beta))) \\ \equiv ? [Mf \alpha] \varphi \wedge ([Ufs U-i [Mf \alpha] \varphi] \vee ([Uff \beta] \gamma \wedge \neg \text{relation}(\varphi, \beta)))) \\ \equiv ++ [Mf \alpha] \varphi \end{aligned}$$

un but posé est satisfait si l'utilisateur accepte l'action comme appropriée, cette acceptation peut être manifestée soit de façon (a) explicite quand l'utilisateur fait savoir qu'il ne garde plus son intention à l'égard de l'action demandée ; soit de façon (b) implicite quand l'utilisateur demande une nouvelle action qui ne se dérive pas de l'action demandée, c'est-à-dire il n'y a pas de relation entre la nouvelle action demandée et les effets de l'action précédente.

- iv) But *abandonné*, l'abréviation $@Mf \alpha$ désigne l'action *Mf* α comme but abandonné.

$$\begin{aligned} Ms Ui [Mf \alpha] \varphi \wedge (Ufs Ui [M-f \alpha] \varphi) \\ \equiv ? [Mf \alpha] \varphi \wedge (Ufs Ui [M-f \alpha] \varphi) \\ \equiv @ [Mf \alpha] \varphi \end{aligned}$$

un but posé est abandonné quand l'utilisateur fait savoir qu'il veut que la machine ne fasse

pas l'action concernée.

Pour le cas où l'utilisateur pose une question *Uffs*, on a le même état-but. La seule différence est qu'on attend une réponse du type *Mfs*². Il faut rappeler que *ffs* est une abréviation pour *ff partager* où *partager* est une action de base. Par conséquent, une question *posée* par l'utilisateur devient une demande de partager des informations :

$$\begin{aligned}
& [Uffs \phi] (Ms Ui [Mfs \phi] \phi) \\
& \equiv [Uff partager \phi] (Ms Ui [Mf partager \phi] \phi) \\
& \equiv [Uff partager \phi] (? [Mf partager \phi] \phi) \\
& \equiv [Uffs \phi] (? [Mfs \phi] \phi)
\end{aligned}$$

4.2.2 Sous-but Déclenché par la Machine

Dans ce cadre proposé pour la coopération homme-machine la machine n'a pas d'intention, néanmoins il est possible qu'elle prenne une certaine initiative dans le dialogue vis-à-vis de la tâche demandée. Le premier cas se produit si la machine ignore la tâche à faire (n'a pas la *savoir-faire* nécessaire), ce qui déclenchera une demande d'information dirigée vers l'utilisateur. Le deuxième cas se produit quand la connaissance en machine entre en contradiction avec la proposition affirmée par l'utilisateur : la machine demande alors à l'utilisateur de confirmer ou d'explicitier sa connaissance. Et enfin, le troisième cas se produit quand l'utilisateur enseigne à la machine une nouvelle tâche suite à une méconnaissance de la machine. Dans tous les cas, c'est toujours l'intention de l'utilisateur qui est à l'origine des actes de la machine. On peut seulement dire que les "initiatives" de la machine sont justifiées non par la tâche elle-même mais par des exigences de coopérativité.

Question posé par la machine : une question est posée par la machine quand l'action à réaliser n'est pas entièrement (ou mal) définie. Cette situation déclenche un sous-but pour demander à l'utilisateur l'information manquante. A la différence d'un but posé par l'utilisateur, la résolution d'un sous-but posé par la machine ne compte que deux états : en effet, si à une question posée par la machine survient une réponse, celle-ci est immédiatement acceptée comme sous-but atteint et satisfait puisque la machine est incapable d'émettre une opinion quelconque sur elle. Dans notre logique cela se traduit par :

- i) Sousbut *posé*, l'abréviation *?Ufs φ* désigne l'action *Ufs φ* comme sousbut posé.

$$[Mffs \phi] (Ms Ui [Ufs \phi] \phi) \equiv [Mffs \phi] (? [Ufs \phi] \phi)$$

- ii) Sousbut *atteint* et *satisfait*, l'abréviation *++Ufs φ* désigne l'action *Ufs φ* comme but atteint.

$$\begin{aligned}
& Ms Ui [Ufs \phi] \phi \wedge [Ufs \phi] \phi \\
& \equiv ? [Ufs \phi] \phi \wedge [Ufs \phi] \phi \\
& \equiv ++ [Ufs \phi] \phi
\end{aligned}$$

un sous-but posé est atteint et, par la même raison, satisfait quand la réponse attendue par la machine est aussi vraie.

- iii) Sous-but *abandonné*, l'abréviation *@ Ufs φ* désigne l'action *Ufs φ* comme but abandonné.

$$\begin{aligned}
& Ms Ui [Ufs \phi] \phi \wedge [Ufs [U \neg fs \phi] \phi] \gamma \\
& \equiv ? [Ufs \phi] \phi \wedge [Ufs [U \neg fs \phi] \phi] \gamma \\
& \equiv @ [Ufs \phi] \phi
\end{aligned}$$

un sous-but posé est abandonné quand l'utilisateur fait savoir, de façon explicite, qu'il ne répond pas à la question posée.

But déplacé en acquisition : une autre situation où la machine prend l'initiative et demande l'intervention de l'utilisateur, se produit pour apprendre une tâche. A l'origine de cette situation on trouve une demande de réalisation d'une tâche, mais celle-ci est inconnue par la machine. Ainsi le but initial est déplacé de façon à conduire le dialogue dans un contexte d'acquisition de la tâche. Pour ce cas, on a les implications suivantes :

² Il est toujours possible penser à un ensemble de catégories comme : $Uffs_{si}$ quand l'utilisateur fait une question oui/non ou $Uffs_{ref}$ si l'utilisateur fait une question pour connaître la valeur d'un certain paramètre [5].

- i) But *déplacé en acquisition*, l'abréviation $?\Delta\alpha$ désigne comme but du dialogue l'acquisition de l'action α .

$$[Mff \alpha](Ms Ui [Uf \alpha] Msf \alpha) \equiv [Mff \alpha](?\Delta [Uf \alpha] Msf \alpha)$$

- ii) But *en acquisition satisfait*, l'abréviation $++\Delta\alpha$ désigne l'acquisition de l'action α comme atteint et satisfait.

$$\begin{aligned} Ms Ui [Uf \alpha] Msf \alpha \wedge [Ufs U-i [Uf \alpha] Msf \alpha]\phi \\ \equiv ?\Delta[Uf \alpha] Msf \alpha \wedge [Ufs U-i [Uf \alpha] Msf \alpha]\phi \\ \equiv ++\Delta [Uf \alpha] Msf \alpha \end{aligned}$$

un but en acquisition est atteint et satisfait quand l'utilisateur fait savoir que la tâche est achevée.

- iii) But *en acquisition abandonné*, l'abréviation $@\Delta\alpha$ désigne un but en acquisition abandonné.

$$\begin{aligned} Ms Ui [Uf \alpha] Msf \alpha \wedge [Ufs Ui [U-f \alpha] Msf \alpha]\phi \\ \equiv ?\Delta[Uf \alpha] Msf \alpha \wedge [Ufs Ui [U-f \alpha] Msf \alpha]\phi \\ \equiv @\Delta [Uf \alpha] Msf \alpha \end{aligned}$$

4.3 La Caractérisation de la Coopération

Comme nous l'avons déjà vu ci-dessus, le dialogue coopératif est orienté vers la réalisation d'une tâche pour laquelle la machine est considérée comme un assistant qui collabore à l'accomplissement de cette tâche. Dans notre cas, le dialogue est aussi le moyen pour *faire-savoir* (faire partager le plan) pour faire une nouvelle tâche. Il s'agit là non de communiquer des informations mais de montrer une série d'actions pour arriver à un but. Par conséquent, l'utilisateur est incliné à montrer son savoir-faire et, le système est incliné par une attitude appropriée, à l'acqu]Ms

(A8) $?\Delta [Uf \alpha]Msf \alpha \wedge ++Mf \beta \supset Ms(\beta \in \text{sequence}(\alpha))$

Toute action satisfaite β pendant un dialogue d'acquisition appartient à la séquence pour faire la tâche α .

5. Exemple

L'exemple suivant montre de façon générale la structure d'un dialogue simple où les échanges convergent vers un but, dans ce cas-là "le dessin d'un carré". Cette tâche est connue par la machine, cependant pour l'accomplir, il faut demander la couleur du carré (pour l'instant, on suppose connus tous les autres paramètres de l'action *dessine_carré*). Ainsi, pour cet exemple, la connaissance en machine est :

(C1) $Msf \text{dessiner-carré} \wedge M-s \text{paramètre}(\text{dessiner_carré}, \text{couleur})$

U: *dessine un carré*

[Uff dessiner-carré]

1. [Uff dessiner-carré](?Mf dessiner-carré)
(définition Uff) *but posé*
2. ?(Mffs paramètre(dessiner_carré, couleur); vérifier(++Ufs paramètre(dessiner_carré, couleur) ; Mf dessiner_carré)
1, C1, A6 *sous-but déclenché*
3. [Mffs paramètre(dessiner_carré, couleur)] (vérifier(++Ufs paramètre(dessiner_carré, couleur); Mf dessiner_carré)
2, A5.i

[Mffs paramètre(dessiner_carré, couleur)]

M: *de quelle couleur?*

4. [Mffs paramètre(dessiner_carré, couleur)] (?Ufs paramètre(dessiner_carré, couleur))
(définition Mffs) *sous-but posé*

U : *rouge*

[Ufs paramètre(dessiner_carré, couleur: rouge)]

5. [Ufs paramètre(dessiner_carré, couleur: rouge)] (Ms paramètre(dessiner_carré, couleur: rouge)
(définition Ufs)
6. ++ Ufs paramètre(dessiner_carré, couleur)
4, 5 *sous-but satisfait*
7. [Mf dessiner-carré](forme(Obj1,carré) \wedge taille(Obj1, 50) \wedge position(Obj1,(342, 875)) \wedge couleur(Obj1,rouge)
3, 6, (définition dessiner-carré)

[Mf dessiner-carré]

M: *<dessin du carré>*

8. + Mf dessiner_carré
1, 7 *but atteint*

Cet échange d'informations est nécessaire car il est dû à l'ignorance de la machine. Pour cela il faut réaliser une série d'actions –en incluant des actes de langage– préalables à la tâche demandée.

6. Conclusion

Le travail présenté met en place les idées de base d'une logique pour le dialogue coopératif. On a caractérisé l'activité de l'utilisateur en fonction de buts à atteindre qui en même temps structurent le dialogue. Cette logique considère l'action comme l'élément central, où une intervention langagière est considérée comme une forme d'action. Ainsi le dialogue conduit la réalisation d'une action et, à la fois, l'action suscite le dialogue ; l'utilisateur montre des actions et la machine apprend ces séquences d'actions ; on exprime donc le but à atteindre à travers l'action ou le plan souhaités.

Cette logique permet de décrire les échanges dialogiques, dans le cadre de la coopération homme-machine, comme le développement du but posé par l'utilisateur. Ce développement est une séquence d'actions (langagières ou non) déterminée par les intentions de l'utilisateur et la connaissance en machine. Il est possible aussi de traiter au cours du dialogue le défaut d'information relative à la tâche à réaliser. Par ailleurs, cette logique introduit un mécanisme de prédiction –p. e. le cas d'une question déclenchée par la machine– qui permet la récupération en cas d'erreur ou incompréhension.

Finalement, il faut préciser qu'un deuxième ensemble d'axiomes est nécessaire pour gérer les différents échanges dialogiques dans les cas (i) de la généralisation d'une tâche, et (ii) de la reconnaissance d'une tâche déjà répertoriée au cours du dialogue. Ces cas sont essentiels pour incorporer une attitude de collaboration envers l'utilisateur. Ces axiomes ont été développés mais ne sont pas décrits dans ce chapitre.

7. Remerciements

Ce travail a été fait avec la contribution de Luis Villaseñor. Je le remercie très vivement.

8. Bibliographie

1. Austin, J.L. *Quand dire, c'est faire*. Edition du Seuil, Paris, 1970.
2. Caelen, J. Vers une logique dialogique. *Séminaire International de Pragmatique*, Jérusalem, 1995.
3. Caelen, J. & Villaseñor, L. Dialogue Homme-Machine et Apprentissage. *Apprentissage par l'interaction*. (ed.) Khaldoun Zreik, p 83-117. Europia Productions. Paris, 1997.
4. Cohen, P. R. & Levesque, H.J. Persistence, Intention and Commitment. *Intentions in Communication*. (eds.) P. R. Cohen, J. Morgan & M. E. Pollack. The MIT Press. 1990.
5. Colineau, N. *Etude des marqueurs discursifs dans le dialogue finalisé*. Thèse Sciences Cognitives, Université Joseph Fourier, Grenoble, 1997.
6. Halpern, J.Y. & Moses, Y. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence* 54, 319-379. Elsevier Science Publishers. 1992.
7. Maybury, M.T. *Intelligent Multimedia Interfaces*. AAAI/MIT Press, USA. 1993
8. Piaget, J. *Development and Learning. Piaget Rediscovered*, Ripple R.E. and Roccastle V.N. ed., School of Education, Cornell University, Ithaca, New York, 1964.
9. Prendinger, H. & Schurz, G. Reasoning about Action and Change. A Dynamic Logic Approach. *Journal of Logic, Language, and Information*, 5:209-245, 1996.
10. Searle, J.R. *Actes de Langage*, Hermann ed., Paris, 1972.
11. Stein, A. & Maier E. Structuring Collaborative Information-Seeking Dialogues. *Knowledge-Based Systems*, Special Issue on Human-Computer Collaboration. Vol.8 March 1995. Elsevier Science B.V.
12. van der Hoek, W., van Linder, B. & Meyer, Ch. J.-J. A Logic of Capabilities (extended abstract). *Proceedings of the Third International Symposium of Logic Foundations of Computer Science*. LFCS, 94. (eds.) A. Nerode & Yu. V. Matiyasevich. St. Petersburg, Russie, juillet 11-14, Springer-Verlag, 1994.
13. van der Hoek, W., van Linder, B. & Meyer, Ch. J.-J. 1994. Unraveling Nondeterminism : On having the Ability to Choose (extended abstract). *Proceedings of the sixth International Conference on Artificial Intelligence : Methodology, Systems, Applications AIMS'94*. (eds.) P. Jorrand & V. Sgurev. Sofia, Bulgarie, septembre 21-24, 1994. World Scientific. 1994.
14. van Linder, B., van der Hoek, W. & Meyer, Ch. J.-J. Communicating Rational Agents. *Proceedings of the 18th German Annual Conference on Artificial Intelligence KI-94 : Advances in Artificial Intelligence*. (eds.) B. Nebel & L. Dreschler-Fischer. Saarbrücken, Allemagne, septembre 18-23, 1994. Springer-Verlag 1994.
15. Vanderveken, D. *La logique illocutoire*. Mandarga éd. Bruxelles, 1990.
16. Vernant, D. Modèle projectif et structure actionnelle du dialogue informatif. *Du dialogue, Recherches sur la philosophie du langage*, Vrin éd., Paris, n°14, p. 295-314, 1992.