

Stratégies pour un système de dialogue oral homme-machine

Sophie Rosset, Samir Bennacef †, Lori Lamel

Groupe Traitement du Langage Parlé

LIMSI-CNRS, BP 133 91403 Orsay Cédex, France

†VECSYS, 3 r. de la Terre de Feu - Les Ulis, 91952 Courtabœuf, France

{rosset, lamel}@limsi.fr

sbennacef@vecsyst.fr

ABSTRACT

In this contribution we report on our methodology for designing and testing different strategies for dialog management, drawing upon our experience with several travel information tasks. In the LIMSI ARISE system for train travel information we have shifted to a 2-level mixed-initiative dialog strategy improving general navigation and error recovering abilities. The revised dialog strategy has resulted in a 5-10% increase (absolute) in dialog success depending upon the word error rate.

1. INTRODUCTION

Obtenir un taux élevé de succès de dialogue en un minimum de temps (i.e. échanges) reste l'un des objectifs les plus importants dans le développement d'un système de dialogue oral homme-machine. Un dialogue bien dirigé par le système permet d'arriver à des taux de succès élevés, mais généralement au prix d'un grand nombre d'échanges système-utilisateur. Un dialogue complètement libre, où l'utilisateur demande ce qu'il veut quand il veut, peut aboutir en très peu d'échanges, mais en cas de problèmes de reconnaissance ou de compréhension le dialogue risque d'échouer complètement par manque de prédiction. Différentes études [Os99, Hua99] ont montré les avantages d'un système à initiative partagée, où l'utilisateur a majoritairement l'initiative du dialogue sauf dans certains cas bien précis comme par exemple des hésitations trop longues de l'utilisateur ou des erreurs repérées par le système. L'initiative partagée permet ainsi une bonne robustesse face aux erreurs, augmente l'importance des demandes de confirmation [Lav99] ainsi que la capacité à gérer les négations et la référence. Les points importants sur lesquels nous avons concentré nos efforts sont donc la navigation et l'initiative de dialogue (i.e. qui de l'utilisateur ou du système doit diriger le dialogue) et la robustesse du système face aux erreurs. Un autre point important concerne la structuration du domaine (le modèle de la tâche) [Ehr99], dont dépend, au moins partiellement, la capacité de navigation du système, i.e. la capacité de délivrer les informations dans un ordre choisi par l'utilisateur et non préétabli par le système.

Le travail décrit ici a été effectué dans le cadre du projet LE-3 ARISE concernant la demande d'informations par téléphone sur les transports SNCF entre plus de 600 gares en France. Les informations à fournir concernent les horaires, les tarifs, les réductions, les réservations et autres prestations. Notre objectif est de permettre un dialogue avec une structure très ouverte, laissant autant que possi-

ble l'utilisateur s'exprimer comme il le souhaite.

Dans la section 2 nous présentons les stratégies mises en œuvre dans notre système de dialogue pour d'une part améliorer la navigation et d'autre part gérer au mieux les erreurs survenant en cours de dialogue. C'est en situation d'erreur que l'initiative doit être reprise par le système. La section 3 décrit quelques expériences et évaluations ayant permis de valider nos choix. Ces évaluations ont été faites, pour la plupart, dans le cadre des évaluations finales du projet ARISE menées par la SNCF (partenaire du projet) auprès du grand public en novembre 1998.

2. STRATÉGIES ET SYSTÈME DE DIALOGUE

Le terme **stratégies** recouvre différents aspects intervenant lors de la conception et de la mise en œuvre d'un système de dialogue oral homme-machine. L'aspect le plus théorique concerne la nature de l'interaction elle-même et reprend quelques-uns des aspects fondamentaux de la communication [Rou91, Aus62]. Il s'agit là de méta-stratégies qui vont avoir une influence sur des choix stratégiques ultérieurs. Le deuxième aspect dépend des choix ergonomiques fixés lors de la conception du système [Ros99]. Nous nous référons ici aux qualités dont nous voulons doter le système. Il s'agit donc de tout ce qui concerne la négociation, la navigation, la souplesse. Le troisième aspect, plus technique, concerne des choix de développement. Ces différents aspects sont fortement liés et dépendants des objectifs généraux fixés au départ, notamment en matière d'ergonomie et sont sous-tendus par les méta-stratégies. Nous rappelons donc nos objectifs pour ensuite décrire comment les atteindre en améliorant la structuration de la tâche et les capacités du gestionnaire de dialogue.

2.1. Les objectifs

Nous voulons d'abord un système de dialogue à initiative partagée permettant à l'utilisateur d'interagir aussi naturellement que possible avec la machine. Ce système doit permettre la navigation, la négociation et doit être capable de détecter et de gérer au mieux les erreurs. Ensuite nous voulons permettre aux utilisateurs confirmés d'aboutir avec un nombre minimal d'échanges.

2.2. Structuration du domaine

La structuration du domaine en tâches ou en fonctionnalités présuppose connaissance et organisation de ce do-

maine (les termes tâche ou fonctionnalité, a peu près synonyme ici, sont utilisés suivant qu'on s'intéresse plutôt au système ou plutôt à l'utilisateur).

Domaine désigne l'ensemble des "secteurs d'activités" possibles, **tâche** les différentes actions possibles côté système et leur organisation, **fonctionnalités** les différentes actions qu'on veut rendre accessibles à l'utilisateur. Pour un domaine donné définir la tâche revient à déterminer ce qu'un utilisateur doit pouvoir faire avec le système. Dans le cas d'ARISE le domaine est celui des trains voyageurs de la SNCF. L'utilisateur doit pouvoir accéder aux informations concernant les horaires des trains incluant les horaires de départ et d'arrivée et les changements le cas échéant. L'utilisateur doit pouvoir effectuer des réservations ou un achat de billets, s'informer sur les réductions ou les tarifs. Après avoir défini les différentes fonctionnalités, la structuration du domaine (le modèle de la tâche) peut être élaborée. Il s'agit de déterminer des relations (fortes ou faibles, hiérarchiques ou non) entre les différentes fonctionnalités, chacune pouvant être découpée en plusieurs buts à atteindre. Ces buts correspondent à des fonctionnalités élémentaires. Afin de rendre le dialogue aussi souple que possible c'est-à-dire de laisser un maximum de liberté à l'utilisateur notamment en matière de navigation et de négociation, nous avons distingué différents types de fonctionnalités (informations horaires, informations services, réservation, achat billet et informations réductions) et nous avons défini des fonctionnalités élémentaires lesquelles ne sont pas nécessairement affiliées à un type de fonctionnalité donnée mais plutôt à l'ensemble de la tâche. Le modèle de la tâche peut être vu comme un réseau dont les nœuds sont des fonctionnalités avec des liens entre ces nœuds si le passage entre fonctionnalités est possible.

2.3. Le gestionnaire de dialogue

Afin d'atteindre nos objectifs le gestionnaire de dialogue, dont le rôle est primordial, doit avoir connaissance de ce qui s'est passé et se passe au cours du dialogue, et surtout il doit être capable de prévoir afin de mieux conseiller ou, le cas échéant diriger l'utilisateur. Ces connaissances sont utilisées différemment selon l'état du dialogue pour lequel différentes phases sont distinguées.

Les connaissances Le gestionnaire dispose d'un ensemble de connaissances statiques (essentiellement connaissances linguistiques et modèle de la tâche) et dynamiques. Ces dernières, que nous décrivons, sont utilisées à différents moments de l'analyse de la requête. Elles participent à l'évaluation de l'état du dialogue. Ces différentes sources de connaissances sont :

1- *Paire d'échange système-utilisateur*: Il s'agit du couple (parole système, parole utilisateur) courant, qui permet d'extraire certaines informations utiles pour la gestion à court terme du dialogue. Par exemple, si à une question du système proposant une réservation, l'utilisateur répond par l'affirmative, la fonctionnalité réservation sera activée.

2- *Historique du dialogue*: Il s'agit d'un historique sur un plus long terme que la simple paire système-utilisateur. Cet historique comprend trois niveaux:

• *Dialogue*: il s'agit de l'histoire du dialogue dans son ensemble, de la liste des tâches¹ ouvertes, et du bon ou

mauvais déroulement du dialogue.

• *Tâche*: il s'agit pour une tâche donnée, de la suite des sous-tâches qui ont été appelées et de l'historique du bon ou mauvais déroulement de la tâche.

• *Sous-tâche*: il s'agit pour une sous-tâche donnée de son histoire c'est à dire de tout ce qui concerne son bon ou mauvais déroulement.

Cet historique va permettre de gérer la navigation, la négociation et le traitement des erreurs.

3- *État du dialogue*: Il est déterminé par la coexistence de certaines informations se rapportant tant à la tâche en cours qu'au déroulement du dialogue (durée, nombre de requêtes, présence d'une incohérence) et par les différentes sources de connaissances. L'état du dialogue permet une adaptation au dialogue en cours, un meilleur repérage des erreurs et une génération de réponse plus adaptée.

Les phases Une phase correspond à l'action qui est en cours dans une (sous-)tâche donnée (p. ex. la négociation dans la recherche d'un horaire). Cela se rapproche à un niveau plus théorique d'un acte de dialogue. Les différentes phases de dialogue identifiées sont l'acquisition, la négociation, la navigation, la post-acceptation et le méta-traitement.

Durant la phase d'**acquisition**, le système doit obtenir les informations nécessaires pour compléter la tâche ouverte. Celles-ci dépendent bien entendu du domaine d'application et du modèle de la tâche. Si l'utilisateur accepte la solution proposée par le système, alors le système entre dans une phase de **post-acceptation**. Dans cette phase, soit l'utilisateur fait une autre demande soit le système propose de lui-même d'autres demandes et ce en fonction du modèle de la tâche qui prend là toute son importance. Cette phase de post-acceptation est une phase de transition pour passer à une phase de **navigation** ou à une phase de **négociation** ou à la clôture du dialogue. La phase de **méta-traitement** concerne le repérage et le traitement des erreurs. Ces phases sont des états à partir desquels le système peut prendre des décisions.

Lors du dialogue, chaque énoncé de l'utilisateur passe par différentes étapes.

1- *Compréhension contextuelle*: il s'agit de l'interprétation du schéma sémantique issu du module de compréhension littérale dans le contexte du dialogue courant.

2- *Décisions*: il s'agit de décider s'il y a un changement de tâche (navigation), une erreur (p.ex. une contradiction entre l'historique du dialogue et l'interprétation contextuelle) ou si le dialogue suit son cours. Le changement de tâche implique une fermeture de la tâche en cours et une activation de la nouvelle tâche. Une trace du changement est conservée de façon à faciliter le retour à la précédente tâche. Si une erreur est détectée, la stratégie pour la traiter dépend de la phase de dialogue dans laquelle se trouve le système (acquisition ou post-acquisition), de la tâche courante, des historiques et du modèle de la tâche.

La navigation La navigation est directement dépendante du modèle de la tâche. Idéalement, nous pourrions envisager la navigation comme une liberté totale laissée à l'utilisateur de changer de fonctionnalité à n'importe quel moment du dialogue. Dans la réalité (pour ARISE en tout cas), le domaine est organisé de telle manière que seuls certains passages sont autorisés. Ceci nous permet d'utiliser au mieux la prédiction et limite les erreurs

décrites dans le cadre du modèle de la tâche.

¹ Les tâches et sous-tâches dont nous parlons maintenant sont l'équivalent système des fonctionnalités et fonctionnalités élémentaires

d'interprétation même lorsqu'elles sont dues au système de reconnaissance. Le module de décision se fonde sur le modèle de la tâche pour décider de la validité de ces passages. Tant que le dialogue se passe à l'intérieur d'une des sous-tâches, telles que nous les avons décrites plus haut, la navigation, sauf cas exceptionnels, n'est pas autorisée. Par exemple, si le système en est encore à acquérir l'un des trois éléments nécessaires à l'accès à la base de données (ville de départ, ville d'arrivée et date de départ), une demande sur les réductions ou les tarifs ne pourra être satisfaite. En revanche le passage d'une sous-tâche *réserve* à *informations réduction* est libre.

Le traitement des erreurs Une erreur représente autant une contradiction due à l'utilisateur (dans une même requête ou entre différentes requêtes) qu'une erreur due au système lui-même. Pour le système, une erreur est surtout une incohérence ou une contradiction dans le temps. Nous distinguons deux catégories d'erreurs: les erreurs systématiques du système de reconnaissance et les autres. Les premières erreurs sont traitées et corrigées lors de la compréhension contextuelle avant même que l'énoncé ne soit analysé par le module de décisions. Ces erreurs sont fortement dépendantes des caractéristiques du système de reconnaissance. Le premier exemple de la figure 1 illustre bien ce type d'erreur: il s'agit ici de la confusion, fréquente, entre "non" et "Nantes". À ce moment-là du dialogue, si le système récupère "Nantes", il le transforme en "non". La nouvelle structuration de la tâche a rendu ce type de traitement possible et relativement performant. La détection d'autres types d'erreur repose sur un contexte plus large que la simple paire système-utilisateur. En cas de conflit détecté lors de la phase d'acquisition, le système passe d'un dialogue libre à un dialogue contraint. Lors du passage à un dialogue contraint le système de reconnaissance bascule vers un lexique et un modèle de langage plus spécifiques. Ce passage est visible pour l'utilisateur dans la mesure où le système lui demande de respecifier l'information contradictoire et lui fournit un exemple de formulation possible, comme le montre l'exemple 2 de la figure 1. Le dialogue contraint peut-être déclenché dans deux cas: le système rencontre une difficulté pour obtenir une des valeurs de base (ville de départ, d'arrivée et date) ou bien lorsque le système rencontre une valeur contradictoire, lors d'une autre phase, toujours à propos d'une des trois valeurs de base. Ce type de contradiction peut aussi bien être due à une erreur ancienne mais non détectée qu'à une erreur récente.

3. EXPÉRIENCES ET ÉVALUATIONS

Un corpus de plus de 3700 appels a été utilisé pour développer et tester les stratégies de dialogue. Le système a été régulièrement évalué par la SNCF (partenaire du projet ARISE). La dernière évaluation a eu lieu en novembre 1998. Les sujets ont été recrutés par une compagnie indépendante. Le taux de réussite était de 15.3% sur les 151 appels évalués.

Nous présentons ici deux résultats d'évaluation portant plus spécifiquement sur la navigation d'une part et d'autre part sur les stratégies de récupération et traitement des erreurs. Pour ces deux évaluations, quatre versions différentes du système ont été testées. Pour l'évaluation de la navigation (voir figure 2), nous avons utilisé deux versions (mai'98 et octobre'98). La première permettait de nav-

S: mercredi 19/04 vous avez un train direct de Paris à Lille à 9h10 arrivant à 10h08. Qu'en pensez-vous? Voulez-vous réserver?
R: je voudrais le tarif
S: Bénéficiez-vous d'une réduction ?
R: Nantes — U: non
S: En quelle classe voulez-vous voyager ?
R: je vais à Lyon
S: Oui, vous allez à Lyon de quelle ville partez-vous ?
R: je vais de Lyon à Lille
S: Je suis désolée, je n'arrive pas à comprendre, indiquez votre ville de départ, par exemple Paris.
R: Lyon
S: Bon, vous partez de Lyon quelle est votre destination ?

Figure 1: Exemples de détection et de traitement d'erreur. S: système; R: reconnaissance; U: utilisateur (si différence avec la reconnaissance)

igner entre certaines fonctionnalités par simple détection de mots-clés dans la requête de l'utilisateur. La deuxième présente une plus grande connectivité entre fonctionnalités, mais les conditions sur les transitions sont plus fortes. Cette version inclut toutes les dernières modifications en matière de structuration de la tâche et de récupération des erreurs. Dans cette version, nous avons implémenté deux niveaux de dialogue (l'un libre et l'autre contraint). Le passage à un niveau contraint se fait lorsque le système détecte des erreurs ou lorsqu'il ne parvient pas à obtenir les informations dont il a besoin pour interroger la base de données. Pour évaluer le traitement des erreurs, nous avons utilisé les données recueillies lors de deux évaluations menées par la SNCF. Le premier système évalué (novembre 97) est la version de base de notre système. Le deuxième est le même que celui utilisé pour les évaluations d'octobre 98 mis à part le nombre de fonctionnalités implémentées. Dans cette version il y a ni réservation ni tarifs, en revanche il y a possibilité de demander des trajet aller-retour.

3.1. La navigation

Nous avons comparé deux versions du système. La première version (mai'98), nous a permis de constater qu'une trop grande liberté de navigation (conditions trop faibles sur les transitions) limitait la capacité de prédiction pourtant nécessaire pour diminuer le risque d'erreurs. Durant l'automne 98, nous avons testé une version dans laquelle la tâche est structurée différemment ce qui permet d'utiliser au mieux la capacité de prédiction, en particulier en limitant les passages d'une tâche à une autre. La figure 2 donne un aperçu partiel des changements opérés dans la structuration du domaine.

Nous avons mesuré les réussites en terme de passage d'une tâche à une autre. Le gain global a été de 10% entre les deux versions (en mai 98: 82% de passages demandés ont réussi et 92% en octobre 98). Le tableau 1 montre le nombre de requêtes avec un changement de tâche et le nombre de fois où le changement portait sur le tarif, la réservation et les services (calculé sur 109 dialogues). Le gain est net en particulier sur la détection de la tâche pour les demandes de tarif.

Il nous fallait également vérifier si le système ne passait pas à une autre tâche alors que celle-ci n'était pas demandée. Entre mai 98 et octobre 98 le nombre de passages

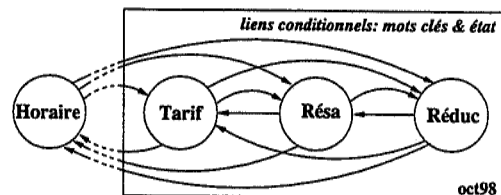
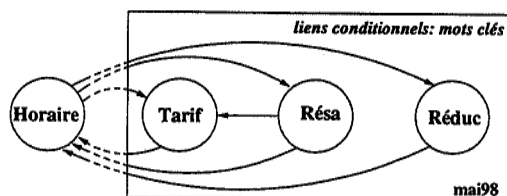


Figure 2: Extraits de la structuration du domaine. La partie gauche correspond au système mai'98, la partie droite au système octobre'98. Plus de possibilités de navigation sont offertes mais avec des conditions sur les arcs dépendant à la fois de la requête de l'utilisateur et de l'état du dialogue.

Table 1: Détection de changements de tâches: # changements demandés / # changements effectués.

Données	Tarif	Résa	Serv.
mai98 (50 dial.)	37/25	67/55	38/37
oct98 (59 dial.)	53/48	61/57	24/22

effectués alors que non demandés a diminué (de 40% des passages effectués qui n'étaient pas demandés en mai 98 à 21% en octobre 98).

3.2. Traitement des erreurs

Une façon simple de vérifier l'apport des stratégies de récupération d'erreurs (notamment celles dues à la reconnaissance, de loin les plus fréquentes) est de mettre en rapport le taux de succès de dialogue et le taux d'erreurs de reconnaissance. Nous avons comparé deux versions du système. La figure 3 nous montre un rapport entre le taux de succès du dialogue et le taux d'erreurs de reconnaissance. Nous constatons qu'à un taux d'erreurs de reconnaissance identique, le taux de réussite de dialogue a augmenté entre les deux versions du système (par exemple, pour 30% d'erreurs de reconnaissance, nous avions en 97 63% de succès dans le dialogue et 73% en 98). Nous pouvons également constater que plus le taux d'erreurs de reconnaissance est élevé plus le gain sur le taux de succès de dialogue est important. Ceci est dû à l'utilisation d'un dialogue à deux niveaux dont l'un (contraint) prend la relève de l'autre (libre) en cas de problèmes ainsi que la nouvelle structuration de la tâche (à la fois libre mais plus prédictive).

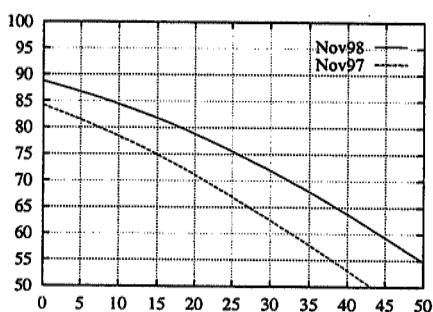


Figure 3: Taux de succès du dialogue en fonction du taux d'erreur de reconnaissance.

4. DISCUSSIONS

Les stratégies regroupent en fait trois ensembles distincts qui interviennent à des moments différents lors du

développement d'un système de dialogue. Ces trois ensembles recouvrent des contraintes méta-stratégiques, des contraintes ergonomiques et des contraintes techniques. Chaque décision prise à l'un de ces trois niveaux va influencer les choix possibles aux autres niveaux. Afin de parvenir à un dialogue peu contraint et permettant une navigation et une négociation aisée, nous avons amélioré la structuration du modèle de la tâche. Toutefois, il s'agit de garantir simultanément une grande liberté à l'utilisateur dans le dialogue et une bonne robustesse face aux erreurs. Laisser l'initiative de la conduite du dialogue majoritairement à l'utilisateur suppose alors une bonne capacité de prédiction sur les choix les plus probables de l'utilisateur. Pour cela, nous avons modifié non seulement le modèle de la tâche mais aussi le gestionnaire de dialogue.

Différentes expériences et évaluations nous ont permis de montrer qu'une navigation, une liberté contrôlées un modèle de la tâche prédictif amélioreraient la qualité du dialogue tout en facilitant le traitement des erreurs.

Nous n'avons pas évalué de façon précise toutes les stratégies, notamment pour le traitement des erreurs. Pour le moment nous pouvons seulement dire que nous avons obtenu une amélioration globale. Quelle stratégie apporte le plus? Pour décider de cela d'autres évaluations sont nécessaires.

BIBLIOGRAPHIE

- [Rou91] E. Roulet, A. Auchlin, J. Moeschler, C. Rubattel, M. Schelling, "L'articulation du discours en français contemporain, 3ème édition, Sciences pour la Communication, Peter Lang, Berne, 1991.
- [Aus62] J. L. Austin, "How to do thing with words", Oxford: Clarendon Press, 1962.
- [Ehr99] U. Ehrlich, "Task hierarchies representing sub-dialogs in speech dialog systems", Eurospeech'99.
- [Hua99] Chao Huang et al., "Lodestar: a mandarin spoken dialogue system for travel information retrieval", Eurospeech'99.
- [Lam98] L. Lamel et al., "The LIMSI ARISE system", IVTTA'98 pp. 209-214, Torino, Sept. 1998.
- [Lav99] C. Alexia Lavelle et al., "Confirmation strategies to improve corrections rates in a telephonic inquiry dialogue system", Eurospeech'99.
- [Os99] Els den Os et al., "Overview of the Arise project", Eurospeech'99.
- [Ros99] S. Rosset, S. Bennacef, L. Lamel, "Design strategies for spoken dialog systems", Eurospeech'99.