

①②

DEMANDE DE BREVET D'INVENTION

A1

②② Date de dépôt : 23.12.98.

③③ Priorité :

④③ Date de mise à la disposition du public de la
demande : 30.06.00 Bulletin 00/26.

⑤⑥ Liste des documents cités dans le rapport de
recherche préliminaire : *Se reporter à la fin du
présent fascicule*

⑥③ Références à d'autres documents nationaux
apparentés :

⑦① Demandeur(s) : FRANCE TELECOM Société ano-
nyme — FR.

⑦② Inventeur(s) : SADEK DAVID, BRETIER PHILIPPE et
PANAGET FRANCK.

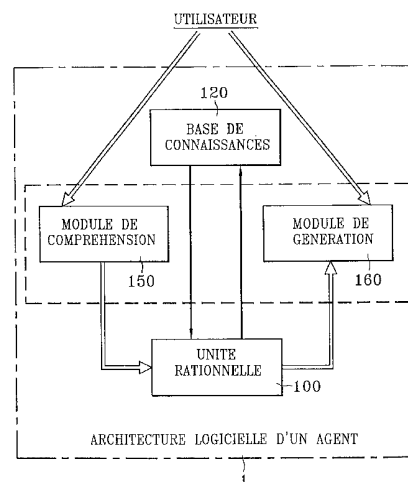
⑦③ Titulaire(s) :

⑦④ Mandataire(s) : CABINET BALLOT SCHMIT.

⑤④ MODELE ET PROCEDE D'IMPLEMENTATION D'UN AGENT RATIONNEL DIALOGUANT, SERVEUR ET
SYSTEME MULTI-AGENT POUR LA MISE EN OEUVRE.

⑤⑦ L'invention concerne un modèle et procédé d'implé-
mentation d'un agent rationnel dialoguant comme noyau
d'un système de dialogue et/ ou comme élément (agent)
d'un système multi-agent comportant les étapes suivantes :

- définition d'une architecture conceptuelle d'un agent
rationnel dialoguant,
- spécification formelle des différents composants de
cette architecture et de leur combinaison permettant d'obte-
nir un modèle formel,
- définition de l'architecture logicielle implémentant l'ar-
chitecture formelle,
- définition des mécanismes de mise en oeuvre des spé-
cifications formelles, l'agent rationnel étant apte ainsi à dia-
loguer avec un autre agent ou avec un utilisateur du
système à travers un quelconque média de communication.



**MODELE ET PROCEDE D'IMPLEMENTATION D'UN AGENT RATIONNEL
DIALOGUANT, SERVEUR ET SYSTEME MULTI-AGENT POUR LA MISE
EN OEUVRE.**

L'invention concerne un modèle et un procédé d'implémentation d'un agent rationnel dialoguant comme noyau d'un système de dialogue ou d'un système multi-agent.

5 L'invention s'applique aux systèmes d'interaction humain-agent (dialogue homme-machine) mais également aux systèmes d'interaction agent-agent (communication et coopération inter-agent).

Elle s'applique à des serveurs d'information.

10 Bien que la conception de systèmes de dialogue homme-machine soit étudiée sérieusement depuis plus d'une trentaine d'année, peu de systèmes préfigurant un usage réel sont aujourd'hui disponibles.

15 La plupart des démonstrateurs qui ont été développés montrent au mieux la capacité d'un système à enchaîner quelques échanges simples avec un utilisateur dans une structure stéréotypée (moulée à celle d'une tâche particulière) et un cadre d'application restreint.

20 Ces systèmes se limitent généralement à illustrer telle ou telle caractéristique d'une interaction évoluée comme par exemple, la compréhension par la machine d'énoncés plus ou moins complexes (contextuels en langage naturel oral ou écrit éventuellement
25 combinés à d'autres média de communication) ou dans certains cas assez limités à la production de réponses coopératives.

Ces systèmes sont encore assez loin de remplir toutes les conditions requises pour un usage naturel
30 desdits systèmes comme des "partenaires" dialoguants

conviviaux même dans des cadres applicatifs bien banalisés.

Les raisons de cette situation sont de deux ordres. D'une part la conception de systèmes de dialogue est
5 une entreprise complexe car elle cumule les problèmes relatifs à la conception de systèmes artificiels intelligents et ceux relatifs à la modélisation et à la formalisation de la communication naturelle. Lorsque
10 l'on s'intéresse au dialogue oral, les problèmes liés à la reconnaissance automatique de la parole accroissent cette difficulté.

D'autre part, beaucoup de travaux ont abordé le dialogue comme un phénomène isolé dont il s'agissait d'identifier les manifestations externes afin de les
15 inculquer en tant que telles à un système automatique. Ces travaux ont (délibérément ou non) fait totalement (ou partiellement) l'économie du lien entre le problème du dialogue et celui de l'intelligence du système et donc d'une étude formelle approfondie des fondements
20 cognitifs du dialogue.

On va maintenant rappeler brièvement les approches classiques du dialogue qui ont été développées jusqu'à présent.

Il y a d'abord les approches structurelles qu'elles
25 soient à vocation informatique ou à vocation linguistique. Elles s'intéressent à la détermination a priori d'une structure de l'interaction qui rend compte des régularités des échanges dans un dialogue (dont les plus simples sont les paires adjacentes telles que les
30 questions-réponses, les suggestions-acceptations).

Ces approches font l'hypothèse que cette structure existe et qu'elle est représentable de façon finie et que tous les dialogues ou du moins une large part d'entre-eux peuvent y être circonscrits. Les approches

structurelles considèrent que la cohérence d'un dialogue est intrinsèque à sa structure et se concentre ainsi sur le co-texte (le texte qui accompagne) en faisant plus ou moins directement l'impasse sur le caractère profondément contextuel de la communication. Ces limitations sont sans appel pour ce qui est de l'intérêt des approches structurelles comme base pour des modèles de l'interaction intelligente.

On a également les approches différentielles classiques.

Ces approches dites aussi orientées plan considèrent une intervention en situation de communication non pas seulement comme une collection de signes (par exemple une séquence de mots) mais comme la réalisation observable d'actions communicatives (appelées aussi selon le contexte, actes de langage ou de dialogue) telles que informer, demander, confirmer, s'engager.

Ces approches ont laissé entrevoir un formidable potentiel pour l'étude de la communication et en particulier du dialogue coopératif. Toutefois, elles empruntent des raccourcis (qui les conduisent à faire appel à des compléments empiriques ou structurels qui les fragilisent) et à des représentations d'utilisation des connaissances qui hélas les conduisent souvent à des aberrations.

Le déposant a développé une nouvelle approche reposant sur l'interaction rationnelle ou agent rationnel dialoguant.

Dans cette nouvelle approche, le déposant a cherché d'abord à rendre maximale la convivialité des interactions entre des utilisateurs et des services automatiques.

On pourra se reporter aux publications suivantes faites sur le sujet:

5 *Sadek 91a:* Sadek M.D. Attitudes mentales et interaction rationnelle: vers une théorie formelle de la communication. Thèse de Doctorat Informatique, Université de Rennes I, France, 1991.

10 *Sadek 91b:* D. Sadek Dialogue acts are rational plans. Proceedings ESCA Tutorial and Research Workshop on the Structure of Multimodal Dialogue, Maratea, Italy, 1991.

Sadek 92: Sadek M.D. A study in the logic of intention. Proceedings of the 3rd Conference on Principles of Knowledge Representation and Reasoning (KR'92), pages 462-473, Cambridge, MA, 1992.

15 *Sadek 93:* Sadek M.D. Fondements du dialogue: l'interaction rationnelle. Actes de la 4ème école d'été sur les Traitements des langues naturelles, pages 229-255, Lannion, France, 1993.

20 *Sadek 94a:* Sadek M.D. Attitudes mentales et fondement du comportement coopératif. Pavard, B., editor, Systèmes coopératifs: de la modélisation à la conception, Octares Eds., pages 93-117, 1994.

25 *Sadek 94b:* Sadek M.D. Communication theory = rationality principles + communicative act models. Proceedings of the AAI'94 Workshop on Planning for Interagent Communication, Seattle, WA, 1994.

Sadek 94c: Sadek M.D. Towards a theory of belief reconstruction: application to communication. In (SPECOM94): 251-263.

30 *Sadek et al 94:* Sadek M.D., Ferrieux A., & Cozannet A. Towards an artificial agent as the kernel of a spoken dialogue system: A progress report. Proceedings of the AAI'94 Workshop on Integration of Natural Language and Speech Processing, Seattle, WA, 1994.

Sadek et al 95 : D. Sadek, P. Bretier, V. Cadoret, A. Cozannet, P. Dupont, A. Ferrieux, & F. Panaget : A cooperative spoken dialogue system based on a rational agent model : A first implementation on the AGS application. Proceedings of the ESCA Tutorial and Research Workshop on Spoken Dialogue Systems, Hanstholm, Danemark, 1995.

Sadek et al 96a : Sadek M.D., Ferrieux A., Cozannet A., Bretier P., Panaget F., & Simonin J. Effective human-computer cooperative spoken dialogue: The AGS demonstrator. In (ISSD 96) (and also Proceedings of ICSLP'96, Philadelphia, 1996).

Sadek et al 97 : M.D. Sadek, P. Bretier, & F. Panaget. ARTIMIS : Natural Dialogue Meets Rational Agency. Proceedings 15th International Joint Conference on Artificial Intelligence (IJCAI'97), Nagoya, Japon, pp 1030-1035, 1997.

Bretier 95 : P. Bretier. La communication orale coopérative : contribution à la modélisation logique et à la mise en oeuvre d'un agent rationnel dialoguant. Thèse de Doctorat Informatique, Université de Paris XIII, 1995.

Bretier et al 95 : P. Bretier, F. Panaget, & D. Sadek. Integrating linguistic capabilities into the formal model of a rational agent : Application to cooperative spoken dialogue. Proceedings of the AAAI'95 Fall Symposium on Rational Agency, Cambridge, MA, 1995.

Bretier & Sadek 95 : P. Bretier & D. Sadek. Designing and implementing a theory of rational interaction to be the kernel of a cooperative spoken dialogue system. Proceedings of the AAAI'95 Fall Symposium on Rational Agency, Cambridge, MA, 1995.

La convivialité de l'interaction se manifeste entre autres par la capacité du système à négocier avec

l'utilisateur, par sa capacité à interpréter des demandes en prenant en compte le contexte, par sa capacité à déterminer les intentions sous-entendues de l'utilisateur et à mener avec lui une interaction flexible qui ne suit pas un plan préconçu une fois pour toute.

Un tel système doit être capable également d'apporter à l'utilisateur des solutions qu'il n'avait pas explicitement demandées mais qui sont néanmoins pertinentes.

Il n'existe pas véritablement à l'heure actuelle de système de dialogue intelligent en service pour une application réelle du fait de la complexité de chacune de ces tâches et du fait de la difficulté à rassembler toutes ces caractéristiques afin que l'interaction puisse être qualifiée réellement de conviviale.

La technologie développée par le déposant repose sur le principe de base qui est que: pour qu'un système automatique puisse mener à bien des dialogues intelligents, ce système ne peut pas être simulé par un automate.

Plus précisément la convivialité du dialogue ne peut pas être conçue comme l'habillage extérieur d'un système préexistant : cette convivialité doit au contraire émerger naturellement de l'intelligence du système.

L'objet de la présente invention est la réalisation d'un agent logiciel qui de par sa construction est rationnel. L'adjonction de principes appropriés le rend également communicatif et coopératif.

En outre, la technologie développée par le déposant permet tout aussi bien l'implémentation d'un agent rationnel dialoguant comme noyau d'un système de dialogue que comme agent d'un système multi-agent.

Dans cette deuxième application (système multi-agent), la communication entre de tels agents ne se fait plus alors en utilisant la langue naturelle mais un langage formel (logique) adapté aux capacités d'interaction desdits agents.

L'invention a plus particulièrement pour objet un modèle et un procédé d'implémentation d'un agent rationnel dialoguant comme noyau d'un système de dialogue ou d'un système multi-agent comportant les étapes suivantes :

- définition d'une architecture conceptuelle d'un agent rationnel dialoguant,
- spécification formelle des différents composants de cette architecture et de leur combinaison permettant d'obtenir un modèle formel,
- définition de l'architecture logicielle implémentant l'architecture formelle,
- définition des mécanismes de mise en oeuvre des spécifications formelles, l'agent rationnel étant apte ainsi à dialoguer avec un autre agent ou avec un utilisateur du système à travers un quelconque média de communication (vocal, ou écrit : écran, clavier, souris etc).

Les différents composants du modèle formel sont dans le même cadre formel unifié (théorie logique) et avec le même formalisme. Le caractère générique des mécanismes et des principes donne au modèle une indépendance par rapport à l'application, aux média de communication et à la langue.

La définition des mécanismes de mise en oeuvre est réalisée de manière à obtenir une correspondance directe entre ces mécanismes et ledit modèle.

La spécification formelle des différents composants de l'architecture formelle et de leur combinaison comporte un niveau d'axiomes de rationalité, un niveau d'axiomes de communication, un niveau d'axiomes de coopération.

La définition de l'architecture logicielle implémentant l'architecture formelle comporte : une unité rationnelle comprenant une couche d'implémentation du niveau d'axiomes de rationalité, une couche d'implémentation du niveau d'axiomes de communication, une couche d'implémentation du niveau d'axiomes de coopération, correspondants respectivement aux axiomes du modèle formel.

La définition de l'architecture logicielle implémentant l'architecture formelle comporte en outre :

- un module de génération et un module de compréhension implémentant une couche de niveau langage naturel.

L'unité rationnelle, le module de génération et le module de compréhension implémentent des mécanismes de mise en oeuvre du modèle formel.

Le module de génération est apte à transcrire un énoncé logique produit par l'unité rationnelle en langage naturel pour l'utilisation du système.

Le module de compréhension est apte à interpréter un énoncé de l'utilisateur en un énoncé logique compréhensible de l'unité rationnelle.

L'invention a également pour objet un serveur d'information, comportant des moyens pour mettre en oeuvre un système de dialogue homme-machine dont le noyau repose sur le modèle et le procédé d'implémentation d'un agent rationnel dialoguant tels que définis précédemment.

L'invention concerne également un système multi-agent comportant des agents communicants, chaque agent comportant des moyens pour mettre en oeuvre une interaction, le système comprenant au moins un agent
5 dont le noyau repose sur le modèle et le procédé d'implémentation d'un agent rationnel dialoguant tels que décrit précédemment.

D'autres particularités et avantages de l'invention
10 apparaîtront clairement à la lecture de la description qui est faite ci-après à titre d'exemple non limitatif et en regard des figures sur lesquelles :

- la figure 1 représente l'architecture logicielle d'un agent rationnel dialoguant,
- 15 - la figure 2 représente l'architecture de l'unité rationnelle et de sa base de connaissance,,
- la figure 3 représente de manière plus détaillée l'architecture logicielle d'un agent dialoguant en tant que noyau d'un système de dialogue (notamment oral),
- 20 - la figure 4 représente une architecture montrant un agent rationnel dialoguant en tant que noyau d'un système multi-agent.

On rappelle que l'approche agent rationnel
25 dialoguant qui a été faite par le déposant et qui a fait l'objet de publications est guidée par des principes de rationalité, de communication et de coopération formalisés dans une théorie de l'interaction rationnelle. On pourra se reporter à
30 cette fin aux publications citée précédemment qui portent l'approche "agent rationnel dialoguant".

La définition de l'architecture conceptuelle d'un agent rationnel dialoguant est donnée en annexe de la description. Cette définition a fait l'objet d'une

publication dans "Conseil Scientifique de France
Télécom, Mémento Technique n°8: Interfaces
Intelligentes et Images" Octobre 1996, pp 37-61.

5 On peut se reporter pour la suite au schéma de la
figure 1.

Selon l'invention, le déposant a mis en oeuvre ces
principes au moyen d'une unité rationnelle 100 qui
constitue le noyau de chaque agent et qui détermine ses
réactions aux évènements externes, que ceux-ci soient
10 des sollicitations (requêtes, réponses, confirmations
etc) d'utilisateurs humains ou bien des sollicitations
d'autres agents logiciels (c'est le cas lorsqu'un agent
est le noyau d'un système multi-agent).

L'unité rationnelle 100 est animée par un moteur
15 d'inférence qui automatise les raisonnements selon les
principes de l'interaction rationnelle que le
programmeur de l'agent peut adapter ou enrichir, de
manière déclarative, en fonction de la tâche à
accomplir.

20 A cette fin, comme cela va être précisé dans la
suite, ces raisonnements sont guidés par des schémas
d'axiomes prédéterminés (énumérés en annexes) et entrés
dans l'unité par le programmeur de l'agent de manière
déclarative en fonction de la tâche que doit remplir
25 ledit agent.

La figure 1 illustre le schéma d'une architecture
logicielle d'un agent dans le cas où une telle
architecture est appliquée à la constitution d'un
système de dialogue avec des utilisateurs.

30 La figure 1 représente donc l'architecture d'un
agent en interaction avec un utilisateur, à travers
comme on va le voir, un module de compréhension 150 et
un module de génération 160. Cette architecture
correspond à une première famille d'application

possible qui est l'interaction (conviviale)
utilisateur-service.

Afin de permettre le dialogue avec des
utilisateurs, l'unité rationnelle 100 est reliée à une
5 interface vers l'extérieur 140.

Cette interface comprend donc le module de
compréhension 150 qui reçoit des énoncés en langue
naturelle et interprète ces énoncés en un énoncé
logique qui sert d'entrée à l'unité rationnelle 100.

10 L'interface comprend également le module de
génération 160 qui exprime la réaction de l'unité
rationnelle 100 en un énoncé de langue naturelle à
destination de l'utilisateur.

Dans ce cadre, l'unité rationnelle 100 est l'entité
15 centrale du service à rendre que ce soit la fourniture
de renseignements (horaires de trains, cours de la
bourse, prévisions météo...) les réservations ou achats
ou encore la recherche d'informations à partir du
réseau Internet.

20 Les principes de coopération implantés dans l'unité
rationnelle et les modules de traitement de la langue
naturelle assurent une interaction conviviale avec
l'utilisateur. Cette interaction peut s'effectuer
directement par la parole en intégrant au système de
25 dialogue ainsi formé des modules de reconnaissance et
de synthèse de la parole (non représentés sur cette
figure).

Cependant, l'unité rationnelle 100 peut toute seule
constituer le noyau d'un agent logiciel autonome. Dans
30 ce cadre, cette unité interagit avec d'autres agents
logiciels par le biais d'un langage de communication
entre agents tel que l'"Agent Communication Langage"
(A.C.L. adopté comme standard par le consortium FIPA).

Les services que l'agent peut rendre sont alors par exemple des transactions sur des marchés électroniques, des tâches d'administration de réseaux, des diffusions d'informations.

5 Ces deux formes d'interaction peuvent être combinées de manière qu'après l'interaction en langue naturelle avec un utilisateur, un agent remplisse une tâche quelconque par des interactions en langage ACL avec d'autres agents logiciels répartis sur les réseaux
10 publics ou privés.

On va maintenant détailler les fonctionnalités de l'architecture logicielle de l'unité rationnelle 100, cette architecture détaillée été illustrée par le schéma de la figure 2.

15 Tout d'abord, l'unité rationnelle 100 implémente des principes issus de la théorie de l'interaction rationnelle dont l'objectif est de formaliser et d'automatiser le comportement rationnel d'un agent en situation d'interaction avec d'autres agents ou des
20 utilisateurs de services.

Cette théorie s'appuie sur deux grandes notions : la notion de logique modale d'une part, dont l'objectif est de permettre de représenter des attitudes mentales d'agents autonomes et, la notion d'actes de langage
25 d'autre part, dont l'objectif est de préciser les effets de communication sur les attitudes mentales des agents.

L'apport de la théorie de l'interaction rationnelle est de formaliser ces deux domaines et surtout
30 l'interaction entre-eux.

L'état d'un agent à un moment donné dans un échange communiquant est ainsi caractérisé par un ensemble d'attitudes mentales.

Les attitudes mentales pouvant être représentées sont par exemple la croyance notée habituellement par l'opérateur K et l'intention notée par l'opérateur I.

5 Ces opérateurs sont indicés par l'agent dont il s'agit de représenter l'attitude mentale.

Dans un dialogue avec le système s et l'utilisateur u, Ks désigne l'opérateur de croyance pour le système et Ku ce même opérateur pour l'utilisateur.

10 Les actes de langage pouvant être modélisés sont entre autres des actes d'information et des demandes. La modélisation consiste en un énoncé logique ou langage logique, par exemple:

Ks Iu Fait ($\langle s, \text{informe Si}(u, p) \rangle$)

cet énoncé logique se traduit comme suit :

15 le système s sait (opérateur K) que l'utilisateur u a l'intention (opérateur I) que soit effectué un certain acte communicatif qui est que s informe u si une certaine proposition p est vraie ou fausse,

20 soit en plus court: "s sait que u veut que s l'informe de la véracité de p".

Le langage logique défini permet d'exprimer des principes de comportement généraux qui vont déterminer les réactions de l'unité rationnelle.

25 Un agent s sera coopératif s'il adopte les intentions de l'utilisateur u. Ceci peut s'exprimer ainsi :

Ks Iu $\phi \rightarrow$ Is ϕ

30 De tels schémas d'axiomes d'une portée très générale sont déjà prédéfinis par la théorie de l'interaction et font partie de l'unité rationnelle d'un agent.

Cependant, le programmeur de l'unité rationnelle peut définir de nouveaux schémas plus spécialisés pour une application donnée.

L'ensemble des schémas guide le raisonnement de l'unité rationnelle 100 et donc ses réactions aux sollicitations de l'environnement.

5 Le calcul de ces réactions est effectué par le moteur d'inférence 101.

L'unité rationnelle 100 comporte donc un ensemble de données 102 qui comprend les axiomes du modèle formel de l'agent rationnel dialoguant. Ces données implémentent les couches de rationalité de communication et de coopération de l'agent.

10 Les sollicitations de l'environnement par exemple les demandes des utilisateurs, ou celles d'autres agents logiciels sont transmises à l'unité rationnelle 100 sous la forme d'un énoncé logique ACL de la théorie de l'interaction rationnelle.

15 Le moteur d'inférence 101 est apte à calculer les conséquences de cet énoncé et en particulier les réponses ou demandes de précisions éventuelles à fournir à l'interlocuteur (qu'il s'agisse d'un agent logiciel ou d'un utilisateur humain) mais aussi d'autres actions non communicatives.

20 Concrètement pour un énoncé donné, le moteur d'inférence 101 examine s'il ne dispose pas d'un principe de comportement qui puisse s'appliquer à cet énoncé pour en déduire la ou les conséquences logiques. Cette procédure est alors appliquée à ces nouvelles conséquences jusqu'à épuisement des possibilités.

25 Parmi toutes ces conséquences, le moteur d'inférence 101 isole les actions de communication ou autres qu'il doit effectuer et qui forment alors la réaction de l'agent rationnel.

30 La première étape de la procédure d'inférence est une mise sous forme normale des énoncés traités afin d'assurer que chaque énoncé ne se présente que sous une

seule forme syntaxique donnée pour pouvoir assurer le tri et la comparaison des énoncés.

5 Cette mise sous forme normale permet également d'assurer une première application des principes simples de raisonnement.

La procédure d'inférence consiste ensuite pour chaque énoncé traité à vérifier si cet énoncé correspond à l'un des schémas d'axiomes 102 qui codent les principes de comportement rationnels retenus.

10 Le mécanisme de cette vérification repose principalement sur l'opération d'unification du langage Prolog.

15 L'ensemble de ces schémas d'axiome peut être modifié par le programmeur de l'unité rationnelle 101 qui peut retirer ou ajouter des schémas d'axiomes ou bien modifier ceux déjà existant pour affiner le comportement de l'unité rationnelle.

20 Ces modifications peuvent s'effectuer dynamiquement. Dans ce cas, l'unité rationnelle modifie son comportement au fur et à mesure.

L'ensemble de la procédure d'inférence est contrôlé de manière à ce que l'unité rationnelle n'entre pas dans des raisonnements infinis. La terminaison de cette procédure est ainsi assurée.

25 Le raisonnement de l'unité rationnelle prend appui sur un ensemble de données qui dépendent fortement de l'application recherchée pour l'agent rationnel.

30 Lorsque l'on souhaite qu'un agent fournisse des horaires de train, il faut que soit à sa disposition des données sur les gares et les connexions entre elles ainsi que les notions temporelles.

L'ensemble de ces données est structuré dans une base de connaissances 120 sous la forme d'un réseau sémantique .

Le réseau sémantique 120 permet d'exprimer des notions de classes et de sous-classes, et d'instance de chaque classe. Il définit également la notion de relation entre classes qui s'applique aux différentes instances des classes.

Par exemple, pour une application de type agenda, le réseau sémantique 120 comprendra au moins les classes "personne" (dont les instances seront l'ensemble des personnes connues dans l'agenda) et "fonction" (dont les instances seront les fonctions connues).

Ces deux classes sont en relation *la-fonction-de*. Pour indiquer que la personne Jean est Publicitaire, le réseau sémantique comprend le fait Prolog: *la-fonction-de* (Jean, Publicitaire).

Un accès au réseau sémantique 120 est réalisé à tout moment durant la procédure d'inférence lorsque les conséquences de l'inférence dépendent de la nature des données.

Dans l'application agenda, par exemple, si l'utilisateur demande quel est le métier de Jean, la réponse de l'agent rationnel va dépendre de son interrogation du réseau sémantique 120.

Le réseau sémantique 120 peut disposer également de notions de proximité sémantique qui sont partiellement utiles pour produire des réactions coopératives de l'agent rationnel.

Il s'agit de donner des distances relatives aux différentes instances du réseau sémantique, ces distances sont déterminées suivant l'application lors de la création du réseau sémantique.

Les instances du réseau sémantique 120 se trouvent ainsi projetées dans un espace métrique dont les dimensions sont les différentes relations du réseau.

Par exemple, la fonction publicitaire sera probablement déterminée comme sémantiquement plus proche de la fonction ingénieur marketing que de garagiste.

5 Cette construction permet d'effectuer deux opérations symétriques appelées relâchement (ou relaxation) et restriction de contraintes.

10 Le relâchement de contraintes vise à donner des réponses proches à des requêtes proches de la requête initiale quand la réponse à celle-ci n'existe pas .

15 Si par exemple on demande à l'agenda quels sont les ingénieurs marketing et qu'il n'en existe pas, la procédure d'inférence peut déclencher une étape de relaxation pour pouvoir donner les coordonnées des publicitaires.

20 La restriction, au contraire, vise à chercher comment préciser une demande trop large. S'il y a 500 publicitaires enregistrés dans l'agenda, une étape de restriction donnera la dimension la plus discriminante de ce trop gros ensemble (par exemple la société ou travaille le publicitaire) afin de pouvoir poser une question pertinente pour identifier la demande de l'utilisateur.

25 La figure 2 permet également d'illustrer que l'unité rationnelle 100 d'un agent rationnel comporte une partie générique indépendante de l'application et une partie dépendante de l'application.

30 On peut considérer que les entrées et sorties de l'unité rationnelle 100 sont des énoncés en ACL. La mise en forme sous forme normale de ces énoncés et la procédure d'inférence sont indépendantes de l'application ainsi qu'une majorité des schémas d'axiomes qui guide le comportement du système. Cependant certains d'entre eux sont adaptés ou créés

spécialement pour l'application de même que le réseau sémantique qui contient les données de l'application. Le réseau 120 doit, la plupart du temps, pouvoir répondre à des demandes de restriction et/ou de relaxation de la part du moteur d'inférence 101 comme on va le voir plus en détail dans la suite.

Ce réseau doit dans ce cas, disposer de notions de distance sémantique entre les instances comme cela a été dit.

Le schéma de la figure 3 illustre avec plus de détails l'architecture logicielle d'un agent selon l'invention.

Le module de compréhension de la langue naturelle interprète un énoncé de l'utilisateur en un énoncé logique compréhensible par l'unité rationnelle 100.

Le vocabulaire traité par ce module dépend en partie du service que doit rendre l'agent rationnel. Cette partie dépendante de l'application est principalement présente dans le réseau sémantique 120 de l'unité rationnelle, ce qui explique que le module de compréhension 150 utilise de nombreuses données provenant du réseau sémantique 120.

Le module de compréhension 150 est apte à prendre en compte l'énoncé de l'utilisateur comme une suite de petites structures syntaxiques (le plus souvent des mots) qui vont chacune activer une (ou plusieurs en cas de synonymes) notion(s) issue(s) du réseau sémantique 120.

Le lien entre le vocabulaire d'entrée de l'utilisateur et le réseau sémantique 120 se fait donc au moyen d'une table d'activation des concepts 131 qui indique quelle(s) notion(s) sémantique correspond(ent) aux mots (ou suite de mots) du vocabulaire.

Ces notions activées dépendent en partie de l'application voulue, mais représentent aussi des concepts beaucoup plus généraux comme la négation, les intentions et connaissances de l'utilisateur, l'existence, les cardinalités etc...

Le module de compréhension dispose donc alors d'une liste de concepts activés (voire de plusieurs en cas de synonymes).

Il est apte à les transformer en un énoncé logique formé par un processus de complétion sémantique. Ce processus part de l'hypothèse de connexité sémantique de l'énoncé de l'utilisateur, c'est à dire que les concepts qu'il a évoqués sont en relation les uns avec les autres.

Le module 150 est apte à relier entre eux, par des relations présentes dans le réseau sémantique, y compris en créant si nécessaire de nouveaux concepts. Le processus détermine les notions sous entendues dans l'énoncé de l'utilisateur.

Il est possible d'indiquer que certaines relations sont incompatibles entre elles dans un énoncé de l'utilisateur. On contrôle ainsi les possibilités de recherche du processus de complétion.

La complétion sémantique fait appel à une fonction de pondération 132 qui permet de fixer un poids numérique pour chaque relation du réseau sémantique, représentant la vraisemblance des vocations de cette relation.

De cette manière, le processus de complétion prend en compte une notion de vraisemblance quand il doit déterminer quels sont les concepts sous-entendus par l'utilisateur. Ces poids permettent également d'associer un coût à chaque interprétation possible en cas de synonyme. Ainsi, un seul énoncé sera finalement

retenu par le module de compréhension, celui au coût le plus faible.

5 Pour faciliter la complétion sémantique, il est également possible de spécifier que certains couples concept-relation ou concept-concept sont implicites. Si seul l'un des concepts a été évoqué et même si l'énoncé étudié est connexe, la relation correspondante sera rajoutée car elle est sous-entendue de manière quasi certaine.

10 Par exemple, dans une application donnant les cours de la bourse, l'énoncé "je voudrais le CAC 40" sera complété de manière implicite en "je voudrais le cours du CAC 40".

15 D'autre part, le module de compréhension 150 doit prendre en compte le contexte de l'énoncé de l'utilisateur.

20 Pour cela, il dispose des concepts précédemment évoqués à la fois par l'utilisateur et par l'agent lui-même dans ses réponses à l'utilisateur. Une partie de ceux-ci peuvent donc être utilisés lors du processus de complétion.

Là encore, on indiquera pour toutes les relations du réseau sémantique, s'il est pertinent de les conserver dans le contexte.

25 Le module de compréhension 150 n'utilise pas d'analyseur syntaxique ou grammatical. Ceci lui permet d'interpréter correctement des énoncés syntaxiquement incorrects, ce qui est particulièrement important dans un contexte de dialogue oral (et d'utilisation de reconnaissance vocale), la syntaxe du discours spontané
30 étant plus lâche .

De plus, l'analyse étant faite par petits composants syntaxiques, il n'est pas nécessaire de construire une grammaire qui va essayer de prévoir à

l'avance l'ensemble des énoncés possibles des utilisateurs.

Enfin, la seule partie dépendant de la langue de l'utilisateur est la table reliant le vocabulaire
5 utilisé aux concepts du réseau sémantique.

Les données sémantiques du réseau représentent en effet des notions universelles. Ce point facilite particulièrement le transfert d'une application d'une langue vers une autre langue.

10 Le module de génération 160 accomplit la tâche inverse du module de compréhension. Il est apte à transcrire une séquence d'actes communicatifs produite par l'unité rationnelle 100 en un énoncé de la langue naturelle de l'utilisateur.

15 Le processus de génération opère en deux phases.

La première phase consiste à prendre toutes les décisions quant au choix linguistique qui s'offre pour verbaliser la séquence d'actes communicatifs fournie en entrée du module.

20 Pour cela le générateur 160 utilise entre autre des éléments du contexte du dialogue pour construire l'énoncé le plus adapté à la situation courante.

Ainsi, dans une application agenda, le module 160 devra effectuer un choix entre des formulations
25 équivalentes comme "le numéro de téléphone de Jean est le" ou "le numéro de Jean est le" ou "son numéro est le", "c'est le..." suivant le contexte du dialogue.

L'objectif de cette première phase est de construire une représentation intermédiaire de l'énoncé
30 en utilisant une notion de ressources linguistiques abstraites 133. Une ressource linguistique abstraite représente soit une ressource lexicale 135, par exemple les noms communs, les verbes, les adjectifs,

soit une ressource grammaticale, c'est à dire la structure syntaxique.

La seconde phase utilise cette représentation abstraite pour construire l'énoncé définitif.

5 Il s'agit d'une étape de traitement qui ne requiert que l'application stricte des règles de grammaire. Parmi ces phénomènes on retrouve par exemple la détermination de l'ordre des constituants de l'énoncé, l'accord entre ces constituants et la déclinaison des
10 verbes.

Les modules de compréhension 150 et de génération 160 utilisent comme format d'entrée, respectivement de sortie, des textes écrits.

15 Si l'on souhaite réaliser une interaction vocale avec un agent rationnel, il faut lui adjoindre des modules de reconnaissance et de synthèse de la parole. Le module de reconnaissance 170 inscrit un signal vocal de l'utilisateur en un texte correspondant à l'énoncé prononcé. Ce module 170 est par exemple indispensable
20 quand un agent rationnel est utilisé comme serveur téléphonique : la seule interaction possible est alors vocale.

L'unité rationnelle y compris le réseau sémantique qui modélise les données que cette dernière manipule,
25 forment ensemble le noyau d'un agent logiciel.

Tel quel, cet agent peut communiquer avec d'autres agents logiciels par exemple à travers le réseau.

Les primitives de communication ACL définies par la théorie de l'interaction rationnelle constituent un
30 langage de communication entre agents qui leur permettent de réaliser des interaction non-ambigües.

Les agents formés par une unité rationnelle 100 et leur réseau sémantique 120 sans les composants d'interaction en langue naturelle (modules 140 et 150)

sont particulièrement bien adaptés à l'utilisation du langage de communication ACL entre agents logiciels pour former des systèmes multi-agent tels que représentés sur la figure 4.

5 L'invention a été mise en oeuvre avec une station SUN Ultra1 (muni d'un processeur 166 megahertz) et sur une station SUN Ultra2 (possédant deux processeurs à 64 bits et une fréquence de 300 megahertz).

10 On utilise une mémoire vive dont la taille peut être aux environ de 32 megaoctets minimum. Le temps de réponse maximal du système est de 2 secondes sur la plate-forme Ultra2 et de 5 secondes sur Ultra1. On peut réaliser la connexion avec un réseau numérique au moyen d'une carte réseau numérique à intégration de service
15 RNIS-Basic Rate Interface.

20 Les trois modules qui ont été décrits, de compréhension 150, de génération 160 et l'unité rationnelle 100 ont été implémentés en Prolog (Quintus version 3.3 pour Solaris 2.5). La communication entre les différents modules et les systèmes de reconnaissance et de synthèse de la parole est réalisé par un programme écrit en langage C, un prototype de l'invention a été développé sous Solaris, mais une
25 version ne comprenant pas les modules de reconnaissance et de synthèse de la parole a été portée sous WINDOWS NT 4.0.

ANNEXE

ELEMENTS DU CADRE DE FORMALISATION LOGIQUE:

5 SPECIFICATIONS FORMELLES

Les concepts d'attitudes mentales (croyance, incertitude, intention) et d'action que l'on manipule ici sont formalisés dans le cadre d'une logique modale du premier ordre (cf. la publication: Sadek 91a, 92, 10 pour les détails de cette logique). On introduit brièvement les aspects du formalisme, dont on se sert dans l'exposé qui suit. Dans la suite, les symboles $\neg$, $\wedge$, $\vee$ et $\Rightarrow$ représentent les connecteurs logiques classiques de négation, conjonction, disjonction et 15 implication, et $\forall$ et $\exists$, les quantificateur universel et existentiel ; p représente une formule close (dénnotant une proposition), ϕ , ψ et δ , des schémas de formules, et i et j (parfois h) des variables schématiques dénotant des agents. On note $\models \phi$ le fait que la formule ϕ est 20 valide.

Les attitudes mentales considérées comme sémantiquement primitives, à savoir la croyance, l'incertitude et le choix (ou la préférence) sont formalisées respectivement par les opérateur modaux K , 25 U , et C . Des formules telles que $K(i,p)$, $U(i,p)$, et $C(i,p)$ peuvent être lues respectivement " i croit (ou pense que) p (est vraie)", " i est incertain de (la vérité de) p " et " i désire que p soit actuellement vraie". Le modèle logique adopté pour l'opérateur K 30 rend compte des propriétés intéressantes pour un agent rationnel, telles que la consistance de ses croyances ou sa capacité d'introspection, formellement caractérisées par la validité de schémas logiques tels que

$$\begin{aligned}
&K(i, \phi) \Rightarrow \neg K(i, \neg \phi), \\
&K(i, \phi) \Rightarrow K(i, K(i, \phi)) \text{ et} \\
&\neg K(i, \phi) \Rightarrow K(i, \neg K(i, \phi)).
\end{aligned}$$

Pour ce qui est de l'incertitude, le modèle logique
 5 garantit également la validité de propriétés
 souhaitables comme, par exemple, le fait qu'un agent ne
 peut être incertain de ses propres attitudes mentales
 ($\models \neg U(i, M(i, \phi))$, M appartenant à $(K, \neg K, C, \neg C, U, \neg U$
 etc $\}$). Le modèle logique pour le choix entraîne des
 10 propriétés telles que le fait qu'un agent "assume" les
 conséquences logiques de ses choix

$$(\models C(i, \phi) \wedge K(i, \Rightarrow \psi) \Rightarrow C(i, \psi)),$$

ou qu'un agent ne peut pas ne pas choisir les cours
 d'événements dans lesquels il pense s'y trouver déjà

15 $(\models K(i, \phi) \Rightarrow C(i, \phi)).$

L'attitude d'intention qui elle n'est pas
 sémantiquement primitive, est formalisée par
 l'opérateur I qui est défini (de façon complexe) à
 partir des opérateurs C et K . Une formule telle que
 20 $I(i, p)$ peut être lue " i à l'intention de réaliser p ".
 La définition de l'intention impose à un agent de ne
 chercher à atteindre que ce qu'il pense non déjà
 atteint ($\models I(i, \phi) \Rightarrow K(i, \phi)$), et garantit le fait qu'un
 agent n'a pas l'intention de réaliser les effets de
 25 bord de ses intentions (ainsi "avoir l'intention de se
 connecter sur un réseau et savoir que cela peut
 contribuer à l'encombrer, n'implique pas
 (nécessairement !) avoir l'intention de contribuer à
 encombrer le réseau").

30 Afin de permettre le raisonnement sur l'action, on
 inclut, en plus des objets individuels et des agents,
 des séquences d'événements. Le langage contient des
 termes (en particulier variables $e, e_1, \dots$) qui
 parcourent l'ensemble de ces séquences. Une séquence

peut être formée d'un seul événement (qui peut être l'événement vide). Afin de pouvoir parler de plans complexes, les événements (ou séquences $\alpha_1 : \alpha_2$, ou des choix non déterministes $\alpha a_1 | \alpha a_2$. Les variables schématiques $\alpha, \alpha_1 \alpha_2 \dots$, sont utilisées pour dénoter des expressions d'action. Sont introduits également les opérateurs *Faisable*, *Fait* et *Agent(i, α)* tels que les formules *Faisable*(αp), *Fait*(αp) et *Agent*(i, α) signifient respectivement que (l'action ou l'expression d'action) α peut avoir lieu après quoi p sera vraie, α vient juste d'avoir lieu avant quoi p était vraie, et i dénote l'unique agent des événements apparaissant dans α .

Une propriété fondamentale de la logique proposée est que les agents modélisés sont en parfait accord avec eux-mêmes sur leurs propres attitudes mentales. Formellement, le schéma $\phi \leftrightarrow K(i, \phi)$, où ϕ est gouvernée par un opérateur modal formalisant une attitude mentale de l'agent i , est valide (Sadek 91a, 92).

Les abréviations suivantes sont utilisées, *Vrai* étant la constante promotionnelle toujours vraie:

	$Faisable(a) \equiv Faisable(a, Vrai)$
	$Fait(a) \equiv Fait(a, Vrai)$
25	$Possible(\phi) \equiv (\exists e) Feasible(e, \phi)$
	$Kiff(i, \phi) \equiv K(i, \phi) \vee K(i, \neg \phi)$
	$Kref(i, \iota x \delta(x)) \equiv (\exists y) K(i, \iota x \delta(x) = y) : \text{l'agent } i \text{ connaît le (ou l'objet qui est un) } \delta, \text{ où } \iota \text{ est l'opérateur de description définie (producteur de termes) tel que :}^*$
30	$\phi(\iota x \delta(x)) \equiv \exists y \phi(y) \wedge \delta(y) \wedge \forall z (\delta(z) \Rightarrow z=y)$
	$Uref(i, \iota x \delta(x)) \equiv (\exists y) U(i, \iota x \delta(x) = y)$

PRINCIPES DE RATIONALITE ET MODELE D'ACTION

Deux principes de rationalité établissent le lien entre les intentions d'un agent et ses plans et actions (Sadek 91a, 94b). Le premier principe stipule qu'un agent ne peut avoir l'intention de réaliser une proposition donnée sans avoir par là-même l'intention que soit faite l'une des actions qu'il pense qu'elle a pour effet la proposition en question, et pour laquelle il n'a pas d'objection particulière à ce qu'elle soit faite. Formellement, ceci s'exprime par la validité du schéma suivant :

$$I(i,p) \Rightarrow I(\text{Fait}(a_1|\dots|a_n))$$

où les a_k sont toutes les actions telles que :

- p est l'effet rationnel de a_k (i.e., la raison pour laquelle a_k est planifié) ;
- l'agent i connaît l'action a_k : $Kref(i,a_k)$
- $\neg C(i, \neg \text{Possible}(\text{Fait}(a_k)))$

Le second principe stipule qu'un agent qui a l'intention que soit faite une action donnée, adopte nécessairement l'intention que cette action soit faisable, s'il ne pense pas qu'elle l'est déjà : ce qui s'exprime formellement par la validité du schéma suivant:

$$I(i,\text{Fait}(a_k)) \Rightarrow K(i,\text{Faisable}(a_k) \vee I(i,K(i,\text{Faisable}(a_k))))$$

La solution au problème de l'effet d'une action est directement liée à l'expression même des principes de rationalité. On considère que si l'on ne peut prédire les effets réels d'une action, on peut cependant dire (de façon valide) ce qu'on attend de l'action, autrement dit, la raison pour laquelle on la sélectionne. C'est en fait ce qui est exprimé par le premier principe de rationalité ci-dessus. Cette sémantique de l'effet d'une action, dans le cadre d'un

modèle du comportement rationnel, permet de dépasser le problème de la non-prédictibilité des effets.

A titre d'exemple, voici un modèle simplifié (pour ce qui est de l'expression des pré-conditions) de l'acte communicatif d'informer de la vérité d'une proposition:

$\langle i, \text{Inform}(j, \phi) \rangle$

Précondition : $K(i, \phi) \wedge \neg K(i, K(j, \phi))$

Effet : $K(j, \phi)$

10

Ce modèle est directement axiomatisé au sein de la théorie logique à travers les principes de rationalité ci-dessus, et du schéma suivant (les actions ne sont donc pars manipulées par un processus de planification comme des structures de données, tel que cela est fait dans le cadre de l'approche orientée-plan classique, mais ont une sémantique logique au sein de la théorie elle-même):

20

$K(h, \text{Faisable}(\langle i, \text{Inform}(j, \phi) \rangle)) \Leftrightarrow K(i, \phi) \wedge \neg K(i, K(j, \phi))$

25

Notons que les deux principes ci-dessus spécifient à eux seuls (sans artifice extra-logique) un algorithme de planification, qui produit déductivement des plans d'action par inférence de chaînes causales d'intentions.

FORMALISATION DE QUELQUES PRINCIPES DU COMPORTEMENT COOPERATIF

30

On peut voir (Sadek 91a, 94a) pour une proposition détaillée d'un modèle du comportement coopératif au sein d'une théorie formelle de l'interaction rationnelle.

L'ADOPTION D'INTENTION OU LE PRINCIPE MINIMAL DE COOPERATION

A priori, rien sur le plan strictement rationnel ne contraint un agent à être (un tant soit peu) coopératif et, en particulier, à réagir aux sollicitations d'autrui (comme par exemple, répondre aux questions qui lui sont posées). Cette condition limite, que nous appelons *principe minimal de coopération*, est un cas particulier de la propriété suivante d'adoption d'intention: si un agent i pense qu'un agent j a l'intention de réaliser une propriété p , et que lui-même n'a pas l'intention contraire, alors i va adopter l'intention que j sache (un jour) que p est réalisée. Cette propriété se traduit formellement par la validité du schéma de formule suivant:

$$(K(i, I(j, p)) \wedge \neg I(i, \neg p)) \Rightarrow I(i, K(j, p))$$

Conjointement, les deux propriétés précédentes garantissent qu'un agent va agir sincèrement, et donc coopérer. Par ailleurs, il est important de souligner qu'elles expriment beaucoup plus qu'un principe de coopération minimale. En fait, elles expriment un principe de coopération "tout-court". Elles traduisent le fait que dès lors qu'un agent est au courant des objectifs d'un autre agent, alors il va l'aider à les atteindre, tant que cela ne contredit pas ses propres objectifs.

LA PERTINENCE

La plupart des types remarquables de réponses coopératives se manifestent par la communication d'un supplément d'information par rapport à ce qui a été explicitement demandé. Cependant, la quantité

d'information additionnelle dépend fortement de l'intérêt présumé du demandeur pour cette information, et en particulier, de ses intentions reconnues. La notion d'intérêt est très contextuelle et reste assez
 5 délicate à établir dans le cas général. A l'inverse, il est des informations qui, de toute évidence, ne sont pas pertinentes pour l'interlocuteur: celles par exemple, (supposées) déjà connues de lui. Autrement dit, éviter la redondance est une composante du
 10 comportement coopératif, ce qui peut s'exprimer comme suit, en termes de propriété élémentaire (qui, en réalité, n'est pas primitive mais dérive directement de la définition même du concept d'intention): si un agent i a l'intention de faire savoir à un agent j une
 15 proposition p , alors i doit penser que j ne la sait pas déjà. Ceci se traduit formellement par la validité du schéma suivant:

$$I(i, K(j, p)) \Rightarrow K(i, \neg K(j, p))$$

20

L'AJUSTEMENT DE CROYANCES

Une réponse corrective est engendrée avec l'intention de corriger une croyance de l'interlocuteur, jugée erronée. Cette croyance
 25 constitue généralement un présupposé inféré (par implicature (Grice 75)) à partir de l'acte communicatif reconnu. L'intention en question est générée chez un agent à chaque fois que sa croyance au sujet d'une proposition dont i ne croit pas son interlocuteur
 30 compétent, se trouve en contradiction avec celle de son interlocuteur. Ceci se traduit formellement par la validité du schéma suivant:

$$K(i, (p \wedge K(j, \neg p))) \Rightarrow I(i, K(j, p))$$

REAGIR AUX SOLLICITATIONS

Dans un système communicant, un agent ne peut se résoudre à la non-reconnaissance par un phénomène qu'il a observé.

5 Pour rendre compte de ce caractère, on formule la double propriété suivante: le premier volet de cette propriété stipule qu'à l'issue d'un phénomène qu'un agent perçoit et auquel, soit il ne peut associer d'événement intelligible, soit tout événement qu'il
10 peut lui associer est inacceptable au vu de ses croyances, l'agent va adopter l'intention de savoir ce qui a été réalisé, typiquement en générant une demande de répétition. Le second volet de cette propriété, qui est moins général que le premier, concerne seulement le
15 cas où l'agent ne peut conformément à son état mental, accepter aucun événement réalisable par ce qu'il a observé ; dans ce cas, l'agent va adopter l'intention de faire savoir à l'auteur de l'événement sa désapprobation vis-à-vis de ce qu'il a "compris", ce
20 qui, en termes d'énoncés linguistiques, peut se manifester, par exemple par l'énonciation de ce qui interdit à l'agent d'admettre l'acte en question.

Formellement, les deux volets de cette propriété s'expriment par la validité des deux schémas suivants,
25 les prédicats *Observe*(*i*,*o*) et *Réalise*(*o*,*e*) signifiant respectivement que l'agent *i* vient juste d'observer l'entité observable *o* (tel un énoncé, par exemple), et que l'entité observable *o* est une façon de réaliser l'événement *e*:

30

- (i) $(\exists e) \text{Fait}(e) \wedge \neg \text{Kref}(i, \text{Fait}(e_1)) \Rightarrow I(i, \text{Kref}(i, \text{Fait}(e_2)))$
- (ii) $(\forall o) (\forall e) [\text{Observe}(i, o) \text{ et } \text{Réalise}(o, e) \wedge \text{Agent}(j, e) \text{ et } \neg \text{Kref}(i, \text{Fait}(e_1)) \Rightarrow I(i, K(j, \neg K(i, \text{Fait}(e))))]$

L'HARMONIE AVEC AUTRUI

le comportement d'un agent en univers multi-agent coopératif doit apparaître, dans ses composantes principales, comme une généralisation de son
 5 comportement vis-à-vis de lui-même. (Par exemple, il doit être valide qu'un agent est sincère, cohérent et "coopératif" avec lui-même). Aussi un agent ne doit pas faire en sorte de faire perdre de l'information aux autres agents. En particulier, il ne doit pas
 10 rechercher l'incertitude pour autrui comme une finalité en soi, sauf, éventuellement, s'il pense que c'est la "bonne" attitude à adopter vis-à-vis d'une proposition donnée: cela suppose que lui-même adopte déjà cette attitude. Pour rendre compte de ce comportement, on
 15 propose la propriété suivante:

$$(i) \ C(i, Possible(U(j, \phi))) \Rightarrow \Gamma_1$$

où Γ_1 peut, par exemple, rendre compte du fait que le choix pour un autre agent d'un futur où il est
 20 incertain d'une proposition, impose ce futur seulement comme une étape transitoire vers une situation de connaissance. Formellement Γ_1 peut être:

$$C(i, (\forall e) (Faisable(e, U(j, p)) \Rightarrow (\exists e') Faisable(e; e', Kif(j, p))) \vee U(i, p))$$

25

Une propriété similaire peut être posée à propos de la recherche de l'ignorance pour autrui. Par exemple, un agent i qui désire qu'un agent j ne croie plus (resp. ne soit plus incertain d') une proposition p
 30 donnée, doit lui-même ne pas croire (resp. ne pas être incertain de) p et désirer que j adopte la même attitude qui lui vis-à-vis de p . On propose alors la propriété suivante:

$$(i) C(i, Possible(\neg K(j, \phi))) \Rightarrow \Gamma_2$$

$$(ii) C(i, Possible(\neg U(j, \phi))) \Rightarrow \Gamma_3$$

5 où les conditions Γ_2 et Γ_3 auront une forme similaire
 aux conditions Γ_1 (les schémas proposés (i), (ii) et
 (iii) restant valides si l'opérateur de choix C est
 remplacé par l'opérateur d'intention I). Nous laissons
 ces conditions volontairement incomplètement
 10 spécifiées, car leur expression précise dépend de la
 façon dont on veut que l'agent modélisé se comporte.
 Elles peuvent, par exemple, tout simplement être
 réduites à la constante propositionnelle *Faux*. Quoi
 qu'il en soit, elles n'ont pas d'incidence sur le reste
 15 de la théorie.

Selon ce que l'on choisit de mettre dans les
 conditions Γ_k , on peut valider des schémas tels que

$$\neg I(i, \neg Kif(j, \phi)), \neg I(i, \neg Kref(j, \lambda \phi(x))), I(i, \neg Uif(j, \phi)) \Rightarrow I(i, Kif(j, \phi)),$$

$$20 \quad \text{ou } I(i, \neg Uref(j, \phi(x))) \Rightarrow I(i, Kref(j, \lambda \phi(x))).$$

L'ACCES LOGIQUE AUX "BOITES NOIRES" DE GESTION DES CONTRAINTES DU DOMAINE

25 Les fonctions "boîtes noires" de gestion des
 contraintes du domaine: relâchement, restriction, sur-
 information, sont directement "accessibles" à partir du
 cadre logique formalisant le comportement de l'agent
 rationnel (voir (Bretier 95)). A titre illustratif,
 l'"accès" à la procédure de sur-information se fait à
 30 travers le schéma suivant, où SURINF est une méta-
 prédicat:

$$K(i, (I(i, K(j, p))) \wedge SURINF(p, q)) \Rightarrow I(i, K(j, q))$$

5 Ce schéma exprime la propriété suivante: si un agent i a l'intention qu'un agent j croie une proposition p et que i pense (par sa fonction de sur-information) que la proposition q peut être une sur-information pertinente de p , alors i va adopter l'intention que j vienne à croire également la proposition q .

REVENDICATIONS

1. Système de dialogue intelligent comprenant au moins un agent rationnel dialoguant comme noyau, caractérisé en ce que ledit agent a été défini par:

- une architecture conceptuelle,
- 5 -une spécification formelle des différents composants de cette architecture et de leur combinaison permettant d'obtenir un modèle formel,
- une architecture logicielle implémentant l'architecture formelle et comprenant unité rationnelle
- 10 (100) apte à dialoguer directement avec l'unité rationnelle d'un autre agent en langage logique,
- et caractérisé en ce que ledit agent est susceptible de comporter en outre deux modules (150,160) implémentant une couche de niveau
- 15 communication en langage naturel, l'un étant un module de compréhension (150) d'un énoncé de langue naturelle et de transcription en langage logique pour l'unité rationnelle (100), et l'autre étant un module de
- 20 génération (160) d'un énoncé de langue naturelle à partir de l'énoncé logique reçu de l'unité rationnelle (100), ces modules étant reliés respectivement à un module de reconnaissance de la parole (170) et à un module de synthèse de la parole (180) pour dialoguer avec un utilisateur.

25

2. Système de dialogue intelligent selon la revendication 1, caractérisé en ce que l'unité rationnelle comprend un moteur d'inférence (101) et des

données (102), ces données comprenant des axiomes du modèle formel, une couche d'implémentation du niveau d'axiomes de rationalité, une couche d'implémentation du niveau d'axiomes de communication, et une couche
5 d'implémentation du niveau d'axiomes de coopération, correspondants respectivement aux axiomes du modèle formel.

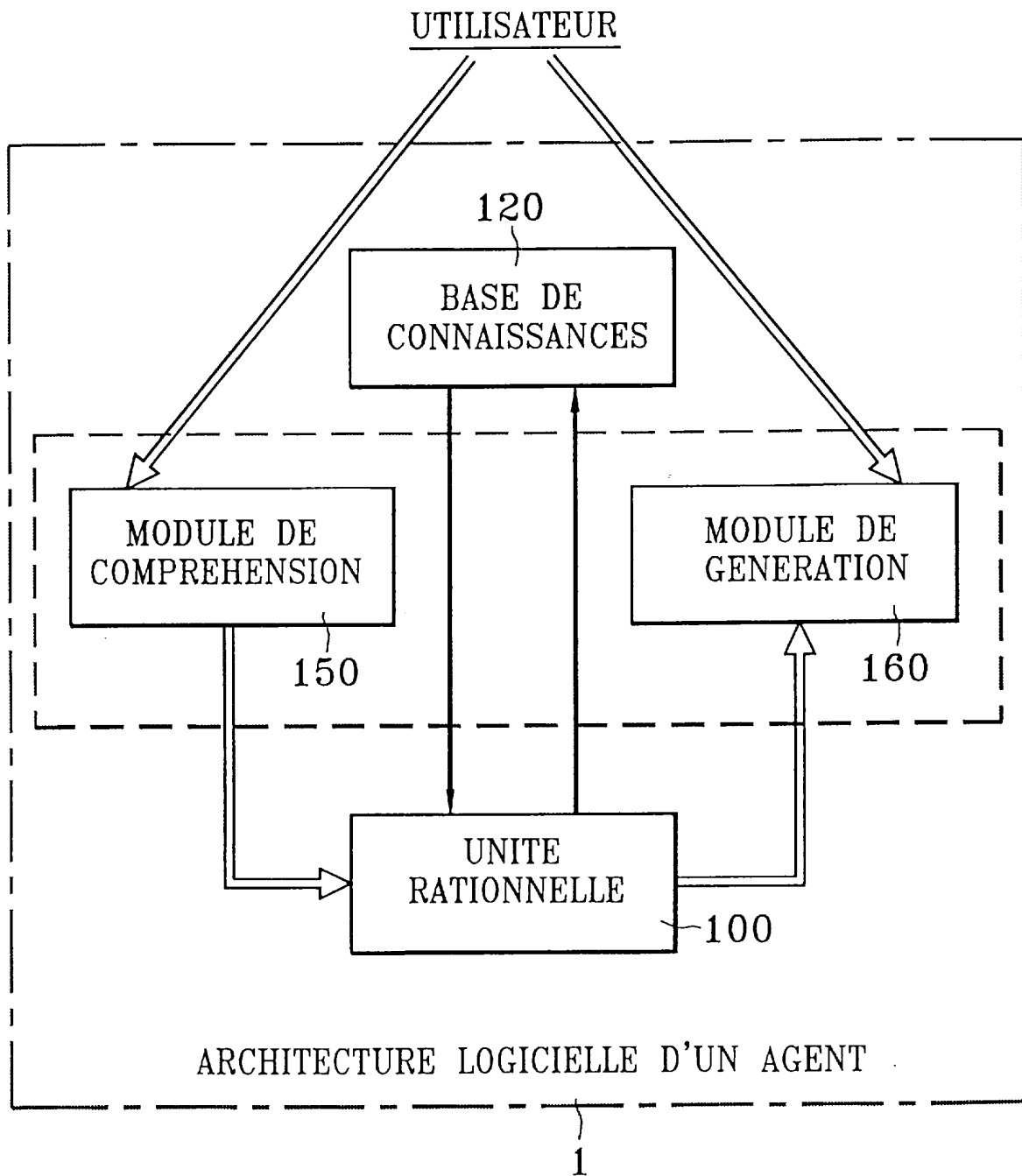
3. Système de dialogue intelligent selon la
10 revendication 1, caractérisé en ce que l'unité rationnelle (100), le module de génération (160) et le module de compréhension (150) implémentent des mécanismes de mise en oeuvre du modèle formel.

15 4. Système multi-agent comportant des agents communicants selon l'une quelconque des revendications précédentes, caractérisé en ce que chaque agent comporte au moins une unité rationnelle (100) pour communiquer avec d'autres agents au moyen d'énoncés
20 logiques, les agents destinés à communiquer avec un utilisateur du système à travers un média de communication comprenant en outre chacun les modules (150,160) implémentant la couche de communication en langage naturel.

25 5. Serveur d'information, caractérisé en ce qu'il comporte des moyens pour mettre en oeuvre un système de dialogue homme-machine dont le noyau comprend un agent rationnel dialoguant selon l'une quelconque des
30 revendications précédentes.

1/4

FIG. 1



2/4

FIG. 2

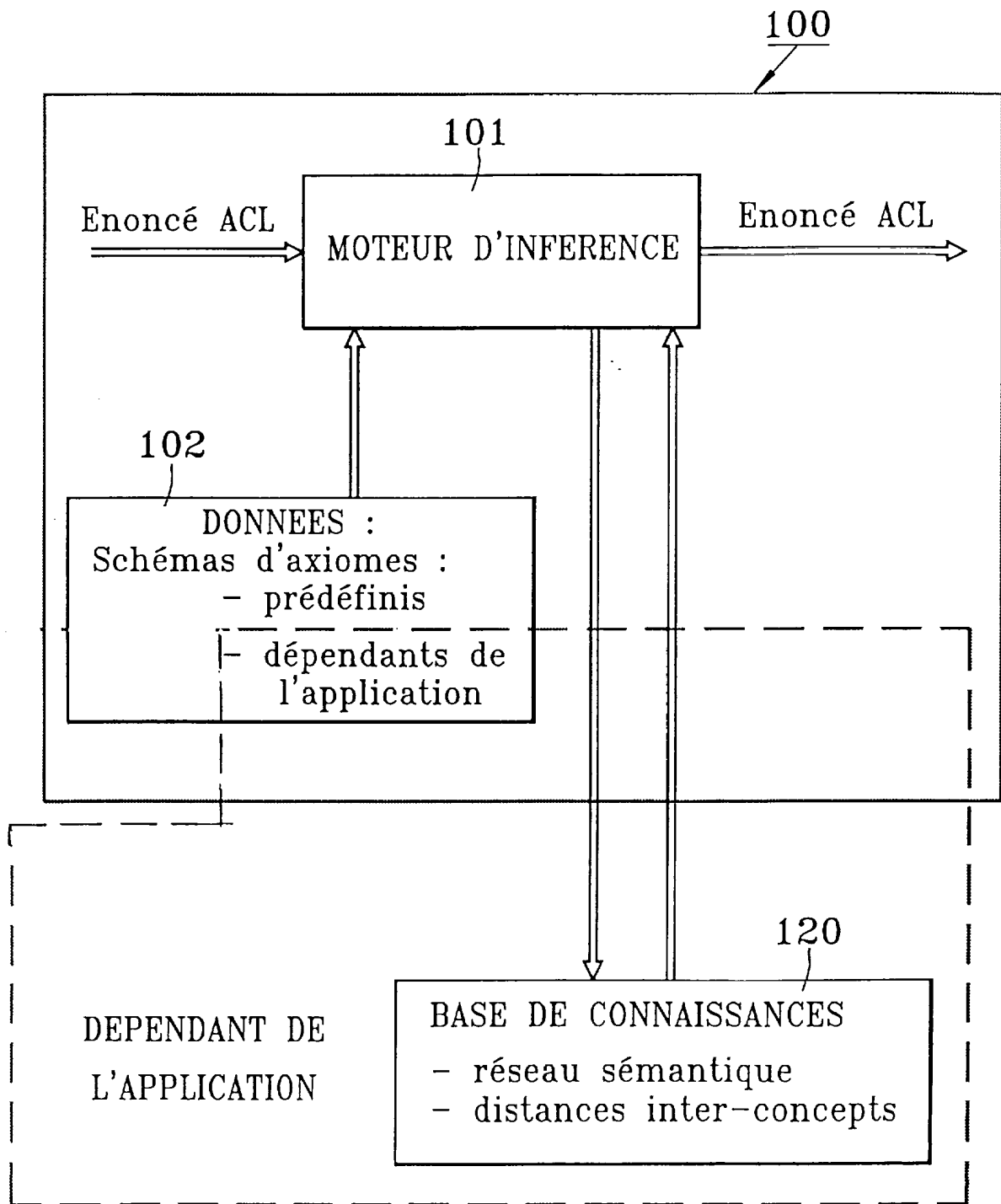
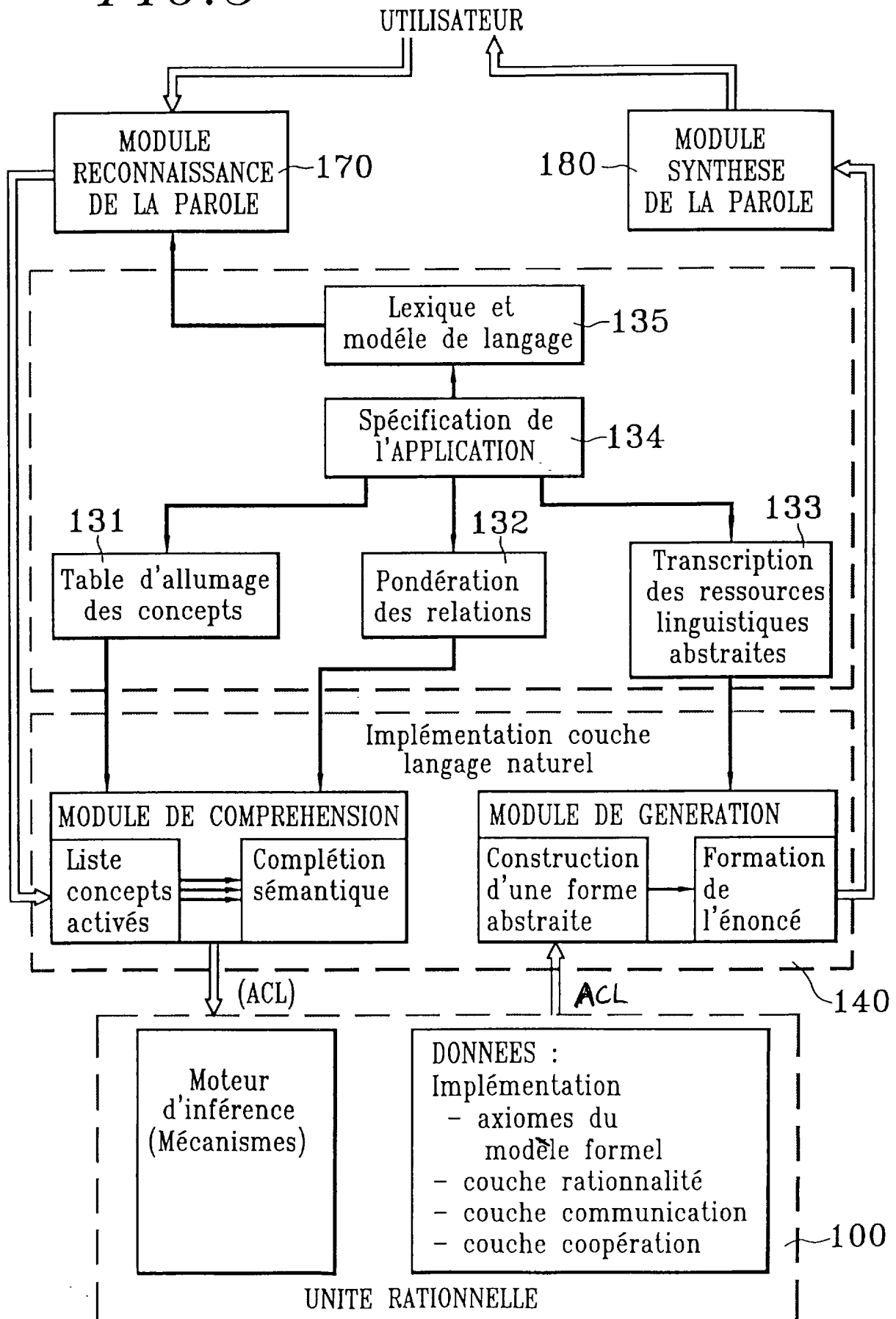
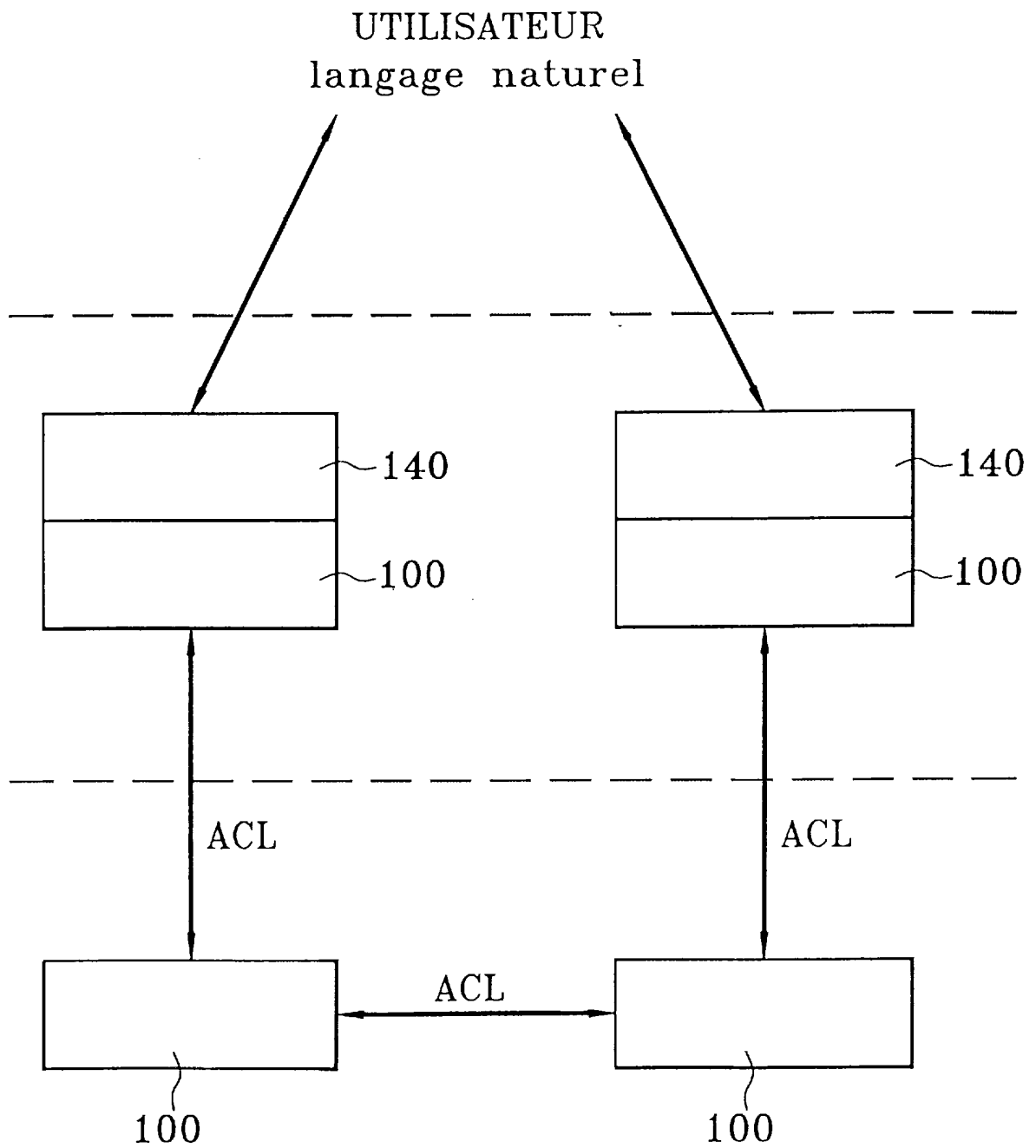


FIG. 3 3/4



4/4

FIG. 4



INSTITUT NATIONAL
de la
PROPRIETE INDUSTRIELLE

**RAPPORT DE RECHERCHE
PRELIMINAIRE**

établi sur la base des dernières revendications
déposées avant le commencement de la recherche

N° d'enregistrement
national

FA 573070
FR 9816374

DOCUMENTS CONSIDERES COMME PERTINENTS		Revendications concernées de la demande examinée
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	
X	LEAVITT J R R ET AL: "THE DIBBS BLACKBOARD CONTROL ARCHITECTURE AND ITS APPLICATION TO DISTRIBUTED NATURAL LANGUAGE PROCESSING" , PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON TOOLS FOR ARTIFICIAL INTELLIGENCE, HERNDON, NOV. 6 - 9, 1990, NR. CONF. 2, PAGE(S) 202 - 208 , INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS XP000236115ISBN: 0-8186-2084-6 * le document en entier *	1,3-5
D,X	SADEK M D ET AL: "Effective human-computer cooperative spoken dialogue: the AGS demonstrator" , PROCEEDINGS ICSLP 96. FOURTH INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING (CAT. NO.96TH8206), PROCEEDING OF FOURTH INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING. ICSLP '96, PHILADELPHIA, PA, USA, 3-6 OCT. 1996 , 1996, NEW YORK, NY, USA, IEEE, USA, PAGE(S) 546 - 549 VOL.1 XP002115169ISBN: 0-7803-3555-4 * le document en entier *	1,3-5
A	WONG S T C: "PREFERENCE-BASED DECISION MAKING FOR COOPERATIVE KNOWLEDGE-BASED SYSTEMS" ACM TRANSACTIONS ON INFORMATION SYSTEMS, vol. 12, no. 4, 1 octobre 1994 (1994-10-01), pages 407-435, XP000498551 ISSN: 1046-8188 * page 414, ligne 10 - page 420, ligne 34 * --- -/--	2
		DOMAINES TECHNIQUES RECHERCHES (Int.CL.6)
		G06F
Date d'achèvement de la recherche		Examineur
13 septembre 1999		Kingma, Y
CATEGORIE DES DOCUMENTS CITES X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général O : divulgation non-écrite P : document intercalaire T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant		

2
EPO FORM 1503 03.82 (P04C13)

INSTITUT NATIONAL
de la
PROPRIETE INDUSTRIELLE

**RAPPORT DE RECHERCHE
PRELIMINAIRE**
établi sur la base des dernières revendications
déposées avant le commencement de la recherche

N° d'enregistrement
national

FA 573070
FR 9816374

DOCUMENTS CONSIDERES COMME PERTINENTS		Revendications concernées de la demande examinée
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	
A	EP 0 689 147 A (CANON KK) 27 décembre 1995 (1995-12-27) * abrégé; figures 2,10 * * page 2, ligne 1 - page 3, colonne 11 * -----	1,3-5
		DOMAINES TECHNIQUES RECHERCHES (Int.CL.6)
Date d'achèvement de la recherche		Examineur
13 septembre 1999		Kingma, Y
CATEGORIE DES DOCUMENTS CITES X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général O : divulgation non-écrite P : document intercalaire T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant		

2

EPO FORM 1503 03.82 (p04C13)