

Un Modèle Cognitif de l'Interaction pour Agents Logiciels

A. Pauchet* ap@lipn.univ-paris13.fr N. Chaignaud† chaignaud@insa-rouen.fr A. El Fallah Seghrouchni‡ Amal.Elfallah@lip6.fr

*LIPN - UMR CNRS 7030
Institut Galilée - Université Paris-Nord, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse

†PSI - FRE CNRS 2645
PSI - INSA Rouen, Place Émile Blondel - BP 08, 76131 Mont-Saint-Aignan

‡LIP6 - UMR CNRS 7606
LIP6 - Université Paris VI, 8 rue du capitaine Scott, 75015 Paris

Résumé :

Cet article présente un modèle d'interaction humaine fondé sur une expérimentation psychologique, ainsi qu'une architecture appelée BDIGGY reliant de façon homogène interaction et planification grâce au concept BDI. BDIGGY comprend un modèle d'interaction humaine conçu sur deux niveaux : 1) un modèle de l'énoncé s'appuyant sur la théorie des actes de langage ; 2) un modèle du discours, sous la forme d'automates temporisés, qui permet de tenir compte de la dynamique des conversations humaines. Les niveaux du discours et des énoncés sont liés par une sémantique des performatives.

Mots-clés : Modélisation cognitive, interaction humaine, résolution coopérative de problèmes, agents BDI, sémantique des actes de langage, automates temporisés.

Abstract:

This article deals with a human model of interaction based on a psychological experiment. It presents an original architecture called BDIGGY where interaction and planning are linked homogeneously thanks to the BDI concept. BDIGGY includes a model of human interaction which is twofold : 1) it is based on the Speech Act theory to model the utterances ; 2) it uses a discourse model, represented by timed automata, to describe the dynamics of human conversations. The utterance level and the discourse level are linked by a semantics of the performatives.

Keywords: Cognitive modelling, human interaction, cooperative problem solving, BDI agents, semantics of speech acts, timed automata.

1 Introduction

De nos jours, la plupart des systèmes informatiques interagissent fréquemment avec les humains. Les humains étant déjà dotés de moyens robustes de communication et de raisonnement, les ordinateurs doivent s'adapter à leurs comportements et offrir des interfaces homme-machine efficaces et agréables. Nous pensons que l'étude, la modélisation et la simulation des interactions humaines peut permettre d'améliorer les capacités d'interaction des agents logiciels avec les humains.

L'étude présentée dans cet article s'appuie sur une expérimentation psychologique [6] durant laquelle des sujets humains avaient à résoudre

un problème de planification en connaissances incomplètes. Un sujet ne pouvait obtenir les informations manquantes qu'en interagissant avec les autres sujets. L'expérimentation a montré que les sujets entrelaçaient planification et interaction durant la résolution du problème.

Cet article propose un modèle de l'interaction humaine et une architecture d'agent, appelée BDIGGY. Cette dernière intègre de manière homogène planification et interaction. Parmi les modèles d'agent existants, les BDI (Deliefs, Desires, Intentions) offrent un cadre intéressant pour développer des agents délibératifs capables de planifier et interagir selon leurs états mentaux. L'architecture BDIGGY intègre le système IGGY étendu à la résolution coopérative de problème [6] et une architecture BDI.

Cet article est organisé de la façon suivante : la section 2 présente le cadre expérimental. La section 3 décrit l'architecture BDIGGY mêlant planification et interaction. La section 4 est consacrée à l'analyse des protocoles expérimentaux. La section 5 détaille le modèle de l'interaction humaine intégré dans BDIGGY. La section 6 situe notre approche par rapport aux travaux existants et la section 7 conclut cet article.

2 Cadre Expérimental

Le problème soumis aux sujets¹ est relatif à une agence de voyage. Trois employés d'une agence ont des compétences complémentaires : le premier est spécialisé dans les transports aériens, le second dans les transports ferroviaires et le dernier dans les transports routiers (bus et taxis). Chacun d'eux est chargé d'organiser un voyage pour un client (une ville de départ et une ville d'arrivée en France, une date et une

¹Le terme *sujet* désigne les sujets humains ayant participé à l'expérimentation psychologique par opposition au terme *agent* qui désigne les agents logiciels implémentés.

heure de départ, une date et une heure d'arrivée, un nombre de voyageurs et un budget). Aucun de ces voyages ne peut se faire *via* un moyen unique de transport. Chaque agent possède son propre problème à résoudre et participe à la résolution des deux autres problèmes. Pour leurs communications, les sujets utilisent un système de courriers électroniques en langage naturel.

L'expérimentation psychologique a permis d'obtenir un échantillon de protocoles expérimentaux (enregistrement des actions, courriers électroniques et verbalisations des sujets) qui ont été analysés du point de vue de la planification [6]. Le modèle de planification humaine résultant a été implémenté et validé dans le système IGGY qui génère, à chaque étape du problème, un *épisode* à partir de la description de l'environnement. Un épisode est la description abstraite d'une classe de plans.

3 L'Architecture BDIGGY

L'architecture BDIGGY est fondée à la fois sur le système IGGY étendu au cadre multi-agents et sur une architecture BDI.

3.1 Les Architectures BDI Classiques

Les agents BDI ont été introduits par Bratman [5], mais il existe de nombreux systèmes BDI, comme celui conçu par Rao et Georgeff [15]. Toute future description du modèle BDI fait référence à dMARS [9]².

Un agent BDI englobe une *file d'événements* stockant les événements internes et externes du système, des *croyances* (connaissances de l'agent), une *librairie de plans* (savoir-faire de l'agent), une pile de *désirs* (buts de l'agent) et une pile d'*intentions* (plans instanciés pour atteindre les buts). Le cycle de l'interpréteur BDI commence par actualiser la file d'événements et les croyances de l'agent. Il active alors de nouveaux désirs en sélectionnant les plans de la librairie de plans qui coïncident avec les événements. Un de ces plans est choisi pour exécution et une fois instancié, il est placé dans la pile des intentions. La première action de la pile d'intentions est exécutée, et ainsi de suite.

3.2 Description de BDIGGY

BDIGGY (voir figure 1) comprend :

- un *module de perception* qui analyse l'environnement et génère des croyances ;
- un *module de planification humaine* (IGGY) qui construit des plans abstraits de façon opportuniste ;
- un *interpréteur de plans* qui se décompose en un *générateur de désirs* interprétant un plan abstrait en désirs et encapsulant une représentation des plans courants, un *générateur d'intentions* raffinant un désir en intentions et un *module d'exécution* effectuant les actions nécessaires à une intention ;
- un *module de communication* qui permet à l'agent d'interagir ; il est appelé par le module de perception pour interpréter les messages reçus et par le module d'exécution pour générer les messages à envoyer.

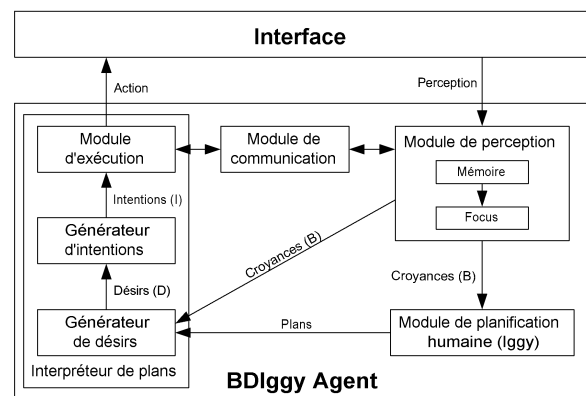


FIG. 1 – L'architecture BDIGGY

Le module de perception contient la *mémoire* de l'agent qui comprend une description du problème (personnalité du sujet à simuler, voyage à construire, carte et connaissances implicites) et une représentation de l'interface (données dynamiques du problème). Le *focus* fait ressortir les croyances pertinentes à un moment de la résolution du problème.

L'interpréteur de plans suit le cycle BDI. S'il y a des actions à accomplir dans la pile d'intentions, le module d'action est activé. Dans le cas contraire, le générateur d'intentions génère de nouvelles intentions en raffinant le désir placé au sommet de la pile des désirs. Si cette pile est vide, le générateur de désirs est appelé pour décider des plans à activer, en fonction des désirs.

3.3 Discussion

Par rapport aux architectures BDI, BDIGGY apporte quelques changements. Tout d'abord, le

²voir [11] pour plus de détails sur les agents BDI.

module de communication permet aux agents d’interagir aisément. Ensuite, la file d’événements est remplacée par le module de perception qui synthétise les changements de l’environnement. Enfin, les plans sont construits dynamiquement par IGGY qui retourne la description abstraite d’une classe de plans.

Les principaux avantages de notre architecture viennent du module de communication pour coopérer avec les autres agents et des plans qui sont générés de façon opportuniste : cela permet de simuler la planification humaine. De plus, la communication et la planification sont représentées ici de manière homogène (par les BDI) dans un même système.

Les sections suivantes se concentrent sur le modèle de l’interaction humaine du module de communication. Ce modèle s’appuie sur l’analyse des protocoles expérimentaux.

4 Analyse de l’Interaction

L’analyse des protocoles expérimentaux se concentre sur l’envoi et la réception de messages, sans se préoccuper des actions locales comme les recherches d’information et les réservations. Cette analyse est faite aux niveaux de l’énoncé et du discours.

4.1 Analyse des Énoncés

Nous avons analysé chaque message individuellement en regardant s’il correspondait à une performative de *Kqml* [12] ou *Fipa-acl* [10], en privilégiant celles de FIPA-ACL. En effet, FIPA-ACL possède une sémantique intéressante [16] pour représenter les états mentaux de l’émetteur. Quand aucune performative ne correspondait au message, nous en avons créée une nouvelle. Les performatives ont été renommées si nécessaire.

Selon la classification de Searle [17]³, les performatives observées proviennent des trois classes suivantes : les *descriptifs* (aussi appelés *assertifs* ou *représentatifs*), les *directifs* et les *engageants*. On notera l’absence de *déclaratif*, ce qui n’est guère surprenant compte tenu du problème qui ne s’y prête pas. De plus, les *expressifs* ne sont que des descriptifs particuliers : la description des sentiments d’un agent est la

description d’une partie du monde, même s’il s’agit d’une introspection.

Le tableau 1 présente les performatives observées dans les dialogues des protocoles expérimentaux. Cette liste est exhaustive dans le cadre des dialogues de recherche d’information.

Descriptifs :	
<i>inform</i>	S envoie spontanément une information à <i>R</i> .
<i>notUnderstood</i>	S ne comprend pas un des précédents messages de <i>R</i> .
<i>reply</i>	S répond à <i>R</i> .
<i>thank</i>	S remercie <i>R</i> .
Directifs :	
<i>acceptProposal</i>	S accepte une proposition d’information de <i>R</i> .
<i>cancel</i>	S annule un de ses précédents messages auprès de <i>R</i> .
<i>query</i>	S demande à <i>R</i> une information.
<i>refine</i>	S demande des précisions sur une des demandes de <i>R</i> .
<i>refuseProposal</i>	S refuse une proposition d’information de <i>R</i> .
Engageants :	
<i>propose</i>	S propose d’envoyer une information à <i>R</i> .

TAB. 1 – Les différents types de performative

La force illocutoire et l’énoncé auquel elle s’applique (notés $F(P)$ dans la théorie des actes de langage) sont étroitement liés, tout comme la performative et son contenu :

- un *descriptif* est appliqué à une *croyance* : il permet au locuteur de décrire la manière dont il perçoit le monde, ce qui correspond à ses croyances ; celles-ci s’ajustent au monde en cas de satisfaction du descriptif ;
- un *directif* s’applique à un *désir du locuteur* : l’émetteur a le désir d’obtenir une information manquante et il transmet ce désir au sujet ayant la connaissance recherchée ;
- un *engageant* s’applique à un *désir de l’interlocuteur* : le locuteur suppose que son interlocuteur recherche une information ; il lui propose ce désir pour qu’il soit confirmé.

4.2 Analyse du Discours

Les interactions ne peuvent être considérées comme une simple succession de questions et de réponses suivant un schéma figé. Chaque locution peut entraîner un grand nombre de comportements différents de l’interlocuteur. L’analyse du discours se fonde sur les travaux de Vanderveken [20], qui étend la théorie des actes de langage au discours. Il maintient la décomposition des conversations en *actes illocutoires*, introduit les *états mentaux* comme unités de base du raisonnement et regroupe en *interventions* les messages dont le thème est identique.

Les interactions des protocoles expérimentaux ne contenaient que des dialogues impliquant

³Cette classification est issue de la théorie des actes de langage introduite par Austin [3] et formalisée par Searle [17]. Une présentation détaillée est faite par Moeschler [13].

deux sujets simultanément. Ces dialogues ont été découpés en interventions. Pour chacune de ces interventions, le sujet initiateur possède une intention discursive caractérisée par le type de sa performative initiale. Les interventions peuvent alors être classées dans l'une des quatre catégories suivantes : *demandes d'information*, *propositions d'information*, *envois spontanés d'information* et *traitements des erreurs*. Les premières et dernières ont une intention discursive primitive de type directif, les secondes de type engageant et les troisièmes de type descriptif. Les traitements des erreurs sont les seules interventions dont l'intention discursive diffère du type de la performative initiatrice puisque celle-ci a été envoyée par erreur. Son intention discursive est de type directif car le receveur du message erroné cherche à faire annuler ce message.

Par ailleurs, la terminaison des interventions définit leur satisfaction. Une intervention peut être satisfaite (réalisation effective du but visé par l'interlocuteur), ou non (réalisation du contraire du but ou non réalisation de ce but).

La performative initiatrice permet de classer l'intervention et la performative de clôture, lorsqu'elle est présente, permet de définir la satisfaction de l'intervention. Cependant, une intervention peut être considérée comme terminée par les interlocuteurs sans qu'il y ait émission explicite d'une performative de terminaison. Seul le cas particulier des envois spontanés d'information ne nécessite qu'un seul acte maître, celui de la performative initiatrice.

Le tableau 2 résume les différentes interventions observées, leur intention discursive primitive et leurs performatives initiatrice et de clôture.

Types d'intervention	Intention discursive	Performative initiatrice	Performative de clôture
Demandes d'information	Directive	query	reply
Propositions d'information	Commissive	propose	reply
Envois spontanés d'information	Descriptive	inform	inform
Traitements des erreurs	Directive	-	cancel

TAB. 2 – Les différents types d'intervention

Tant que l'interlocuteur ne rencontre pas de difficultés d'interprétation d'un énoncé (un non-attendu), il conduit l'intervention courante comme un dialogue *régissant*. Sinon, il initie un dialogue *incident* (une digression dans la théorie de Vanderveken) pour résoudre le problème

apparu, en recherchant une interprétation commune. Quand cette interprétation commune apparaît, il réactive le dialogue régissant mis en suspens. Les dialogues incidents débutent toujours par l'émission d'un *refine*.

Dans le cadre de cette expérimentation, pour laquelle les messages sont des courriers électroniques, le facteur temps est primordial pour les relances de demande et la terminaison des interventions. Par exemple, lors d'une demande d'information, si le locuteur ne reçoit pas de réponse suffisamment vite à son goût, il réitérera sa demande. De même, pour les interventions ne possédant pas de message de clôture, elles sont considérées comme terminées après écoulement d'un certain délai.

4.3 Discussion

Cette analyse de l'interaction est faite aux niveaux des énoncés et du discours. Au niveau des énoncés, chaque message correspond à une performative appliquée à un état mental. Au niveau du discours, tous les messages en relation avec un message initiateur sont regroupés en une seule intervention dont l'intention discursive est donnée par le type du message initiateur.

Les interactions observées présentent à la fois des caractéristiques liées aux dialogues par mails (importance du temps, entrelacement des conversations) et aux dialogues humains (erreurs, imprécisions). Ainsi, KQML et FIPA-ACL, qui utilisent des protocoles de communication rigides, sont inadaptés à la représentation des protocoles expérimentaux.

5 Modélisation des Interactions

Le modèle des interactions humaines respecte l'analyse des protocoles expérimentaux. Il a été intégré au module de communication de BDIGGY. Dans cette section, nous décrivons tout d'abord le problème de l'agence de voyage pour définir les objets manipulés dans les messages. Ensuite, nous représentons les interventions sous forme d'automates temporisés permettant de gérer la dynamique conversationnelle. Enfin, nous donnons une sémantique des performatives utilisées.

5.1 Description Formelle du Problème

Le problème se formalise ainsi :

- CST est l'ensemble des constantes.
- VAR est l'ensemble des variables.
- $TER = CST \cup VAR$ est l'ensemble des termes.
- PRE est l'ensemble des prédicats. Un prédicat débute par "p" et s'applique à une liste (éventuellement vide) de termes et/ou de prédicats. Dans un prédicat, un terme peut être remplacé par "*" s'il n'a aucune importance pour l'agent, par "?" s'il est inconnu, ou par un intervalle.
- FON est l'ensemble des fonctions. Une fonction débute par "f", s'applique à une liste (éventuellement vide) de prédicats et retourne un terme.
- ACT est l'ensemble des actions que l'agent peut accomplir pendant la résolution du problème. Une action débute par "a". Les actions peuvent être internes (leur portée se limite à l'agent) ou externe (via l'interface). Elles peuvent aussi être élémentaires ou abstraites.

Les ensembles de termes suivants sont définis :

- CIT est l'ensemble des villes.
- TIM est l'ensemble des horaires.
- DAT est l'ensemble des dates.
- MEA est l'ensemble des moyens de transport.
- AER est l'ensemble des compagnies aériennes.
- AGT est l'ensemble des agents.

$MEN = BEL \cup DES \cup INT$ est l'ensemble des prédicats sur les états mentaux, où :

- DES est l'ensemble des désirs. Un désir est lié à un agent : $pD(A \delta) \in DES$, avec $A \in AGT$ et $\delta \in PRE$. Ainsi, chaque agent peut raisonner sur les désirs des autres agents concernant les autres problèmes.
- INT est l'ensemble des intentions. Une intention s'applique à une action : $pI(\varphi) \in INT$, avec $\varphi \in ACT$.
- BEL est l'ensemble des croyances. Une croyance s'applique à un prédicat ou à un désir, pour que les agents puissent raisonner sur les désirs des autres agents : $pB(\alpha) \in BEL$, avec $(\alpha \in DES) \vee (\alpha \in PRE \text{ et } \alpha \text{ ne contient aucune donnée non instanciée})$.

Avec ces prédicats, les agents peuvent raisonner sur leurs croyances, leurs intentions, leurs désirs ainsi que sur les désirs des autres agents. Ils ne peuvent par contre raisonner ni sur les croyances, ni sur les intentions des autres agents.

Soit STA l'ensemble des étapes. Une étape est décrite par le prédicat

$$pStage(C_D C_A T_D T_A M N P R)$$

où $C_D \in CIT$ est la ville de départ, $C_A \in CIT$ est la ville d'arrivée, $T_D \in TIM$ est l'heure de départ, $T_A \in TIM$ est l'heure d'arrivée, $M \in MEA$ est le moyen de transport, $N \in \mathbb{N}^*$ est le nombre de voyageurs, $P \in \mathbb{R}$ est le prix total du voyage et $R \in \{true, false\}$ est un booléen dont la valeur vaut *true* si l'étape a été réservée et *false* sinon. Le problème initialement soumis à l'agent est décrit par une étape.

Soit TRA l'ensemble des voyages. Un voyage est représenté par le prédicat

$$pTravel(S_{1,1}[\dots|S_{1,n}] \dots S_{t,1}[\dots|S_{t,m}])$$

où $S_{i,j} \in STA$, $\forall i, j \in \mathbb{N}^*$, sont les étapes du voyage. $\forall k \in \mathbb{N}^*$ fixé, les $S_{k,l}$, $\forall l \in \mathbb{N}^*$ sont les sous-buts parallèles du plan.

Soit PRO l'ensemble des états des problèmes. Un problème est décrit par le prédicat $pProblem(A B)$, avec $A \in AGT$ et $B \in \{true, false\}$. $B = true$ si le problème est résolu et $B = false$ dans le cas contraire. Dans ce prédicat, le problème résolu n'est pas précisé, conformément à ce qui a été observé dans les protocoles expérimentaux.

Soit MES l'ensemble des messages. Un message est décrit par le prédicat

$$pMessage(A_S A_R P O)$$

où $A_S \in AGT$ est l'émetteur, $A_R \in AGT$ est le receveur, $P \in PER$ est la performativité et $O \in MEN$ est un état mental auquel s'applique la performativité. Comme souligné section 4, O est lié à P et ainsi un descriptif s'applique à une croyance, un directif à un désir de l'émetteur et un engageant à un désir du receveur. Par exemple :

- Si A désire que B lui envoie des horaires de trains entre Angers et Paris après 08 :00, il enverra le message : $pMessage(A B \text{ query } pD(A \text{ pStage(Angers Paris (> 08 :00)) ? train ? ? false}))$.
- Ensuite, si B désire répondre à A qu'il y a un train à 09 :00 entre Angers et Paris, il enverra le message : $pMessage(B A \text{ reply } pB(pStage(Angers Paris 09 :00 10 :30 train 1 100 false)))$.
- Enfin, si B désire proposer à A de lui envoyer les horaires du train Angers-Paris après 09 :00, il enverra le message : $pMessage(B A \text{ propose } pD(A \text{ pStage(Angers Paris (> 09 :00)) ? train ? ? false}))$.

Par manque de place, les actions élémentaires externes ne sont pas données ici. Cependant, les agents ont besoin d'actions élémentaires internes pour fonctionner :

- $aAdd(pB)$, avec $pB \in BEL$, ajoute la croyance correspondant à pB aux croyances de l'agent.
- $aDel(pB)$, avec $pB \in BEL$, supprime la croyance correspondant à pB des croyances de l'agent. Cette action est utilisée à la réception d'une performative de type *error*.
- $aDel(pD)$, avec $pD \in DES$, supprime le désir correspondant à pD des désirs de l'agent. Cette action est utilisée à la réception d'une performative de type *error*.
- $aRefine(A S)$, avec $A \in AGT$ et $S \in STA$ est utilisé à la réception d'une performative de type *refine*.

Pour extraire les informations utiles des messages, les agents utilisent des fonctions, comme $fExtract(M)$, avec $M \in MES$, qui extrait l'étape d'un message.

5.2 Modélisation de la Dynamique Conversationnelle des Interactions

Dans ce modèle, une performative s'applique à un état mental, qui porte sur un prédicat. Les états mentaux des agents sont traduits à l'émission d'un message et les messages sont interprétés en états mentaux à leur réception.

Dans le cas particulier de cette application pour laquelle les messages sont des mails, le temps est primordial pour terminer les interventions et gérer les redemandes. Pour modéliser les échanges de messages et leur temporalité, nous utilisons des automates temporisés [2] qui servent à guider l'agent :

- dans la génération des messages : ces derniers sont produits en suivant rigoureusement l'automate. Si un choix se présente, la décision est prise en fonction de l'étape courante et de la personnalité du sujet.
- dans l'interprétation des messages : les automates permettent de gérer les attendus du dialogue. Plusieurs interventions pouvant se dérouler simultanément entre deux mêmes interlocuteurs, ces attendus caractérisent l'appartenance d'un message à une intervention.

Une paire d'automates (un pour chaque interlocuteur) par type d'intervention a été construite. Dans cet article, seule la demande d'information est développée. La figure 2 décrit toutes les

interactions observées dans les protocoles expérimentaux qui suivent une demande d'information. L'automate Q_{ini} correspond au comportement de l'initiateur d'une demande d'information tandis que Q_{int} décrit le comportement de son interlocuteur.

Pour la gestion du temps, Q_{ini} contient une horloge t qui respecte le délai $tsync$ avant que A_{ini} (l'initiateur de l'intervention) ne considère que l'intervention soit terminée. Un compteur m permet de dénombrer le nombre de demandes d'information effectuées avant de recevoir éventuellement une réponse de A_{int} (l'interlocuteur).

Dans Q_{ini} , A_{ini} fait sa demande d'information (état $s1$). S'il reçoit une réponse sous la forme d'un ou plusieurs messages ($s2$), si cette réponse est satisfaisante, il peut ne rien faire ou remercier A_{int} ($s3$). L'intervention est alors satisfaite, même sans clôture explicite. Par contre, si la réponse de A_{int} est incomplète ou inappropriée, A_{ini} abandonne l'intervention explicitement en envoyant un *cancel* à A_{int} ($s6$) ou non, ou reformule sa question ($s1$). A_{int} peut s'être trompé dans sa réponse et A_{ini} sera prévenu par un *cancel* ($s1$). En $s1$, si A_{ini} attend un temps $tsync$ sans réponse de A_{int} ($s4$), il incrémente son compteur m et repose sa question auprès de A_{int} . Si cela arrive trop souvent ($s5$), il peut alors abandonner l'intervention en prévenant A_{int} ($s1$) avec un *cancel*, ou non. Si A_{ini} n'a plus besoin de l'information, il envoie un *cancel* ($s6$). Si A_{int} demande plus de précisions sur la demande d' A_{ini} par un *refine* ($s7$), on entre alors dans un dialogue incident de demande d'information. Ce dialogue peut être satisfait par un *query* et l'intervention courante est réactivée ($s1$), ou peut échouer et A_{ini} doit abandonner sa requête explicitement avec un *cancel* ($s7$) ou non. Parfois, A_{int} satisfait la demande de A_{ini} avant de recevoir des précisions suite à son *refine*. Dans ce cas, A_{int} envoie un *reply* à A_{ini} . En $s1$, A_{int} peut signifier à A_{ini} qu'il ne comprend pas sa demande par un *notUnderstood* ($s8$). L'intervention est alors abandonnée explicitement (par l'envoi d'un *cancel*) ou non par A_{ini} , elle a échoué.

Q_{int} fonctionne de manière similaire à Q_{ini} .

Tous les automates ont été testés sur les protocoles expérimentaux pour vérifier qu'ils couvriraient bien l'intégralité des dialogues. Pour chaque automate, les états et les transitions ont été annotés par leur fréquence d'apparition dans les protocoles. Ces fréquences sont utilisées par exemple lorsqu'un agent souhaite générer un

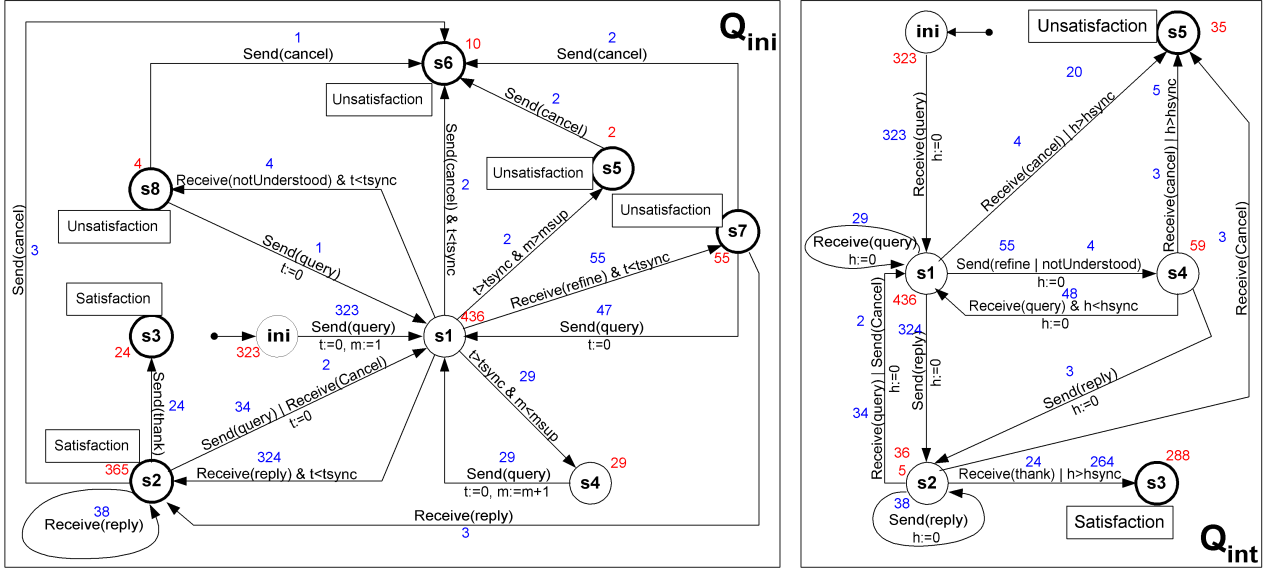


FIG. 2 – Automates de demandes d'information

message. Si l'état possède plusieurs transitions pouvant être franchies, la décision est prise aléatoirement en respectant les fréquences annotées dans l'automate.

5.3 Sémantique des Performatives

Cette section décrit la sémantique associée à l'envoi et à la réception d'un message. Pour chaque performative, la syntaxe du message est donnée ainsi qu'une courte description des actions à effectuer et la sémantique associée, mis sous la forme de la règle générique de réduction

$$\frac{Q_X(s_1, \dots, s_n) \xrightarrow{!/? \text{ per formative}} Q_X(sf)}{a_1; \dots; a_n}$$

où Q_X avec $X \in \{ini, int\}$ est l'automate, $s_1 \dots s_n$ sont les états avant le traitement de l'envoi/réception du message, sf est l'état après traitement, ! indique l'envoi de la performative, ? indique la réception de la performative et $a_1, \dots, a_n \in ACT$ sont les actions à réaliser lors du traitement.

La sémantique indiquée ici ne concerne que les états de Q_{ini} et Q_{int} .

- **acceptProposal**

Syntaxe : $pMessage(A_S A_R \text{ acceptProposal } pD(A_S S))$ avec $A_S, A_R \in AGT$ et $S \in STA$.
Description : A_R ajoute à ses croyances la croyance que A_S désire une information à propos d'une étape, sauf si le message ne lui est pas destiné.

Sémantique : Absent de Q_{ini} et Q_{int} .

- **cancel**

Syntaxe : $pMessage(A_S A_R \text{ cancel } pD(A_S \neg M))$ avec $A_S, A_R \in AGT$ et $M \in MES$.

Description : Si A_S désire annuler un *reply*, alors A_R supprime de ses croyances l'étape contenue dans le message. Si A_S désire annuler un *query*, alors A_R supprime de ses croyances le désir qu'a A_S pour l'étape contenue dans le message.

Sémantique :

$$\frac{Q_{ini}(s_1, s_2, s_5, s_7, s_8) \xrightarrow{!cancel} Q_{ini}(s_6)}{aUpdateTA(M)}$$

$$\frac{Q_{int}(s_2) \xrightarrow{!cancel} Q_{int}(s_1)}{aUpdateTA(M)}$$

$$\frac{Q_{int}(s_1, s_2, s_4) \xrightarrow{?cancel} Q_{int}(s_5)}{aDel(pD(A_S, fExtract(M))); aUpdateTA(M)}$$

$$\frac{Q_{ini}(s_2) \xrightarrow{?cancel} Q_{ini}(s_1)}{aDel(pB(fExtract(M))); aUpdateTA(M)}$$

- **inform**

Syntaxe : $pMessage(A_S A_R \text{ inform } pB(P))$ avec $A_S, A_R \in AGT$ et $P \in PRO$.

Description : A_R ajoute à ses croyances l'information reçue.

Sémantique : Absent de Q_{ini} et Q_{int} .

- **notUnderstood**

Syntaxe : $pMessage(A_S A_R \text{ notUnderstood } pB(M))$ où $A_S, A_R \in AGT$ et $M \in MES$.

Description : A_R ajoute à ses croyances qu'il a l'intention de vérifier le message.

Sémantique :

$$\frac{Q_{int}(s1) \xrightarrow{!notUnderstood} Q_{int}(s4)}{aUpdateTA(M)}$$

- **propose**

Syntaxe : $pMessage(A_S A_R propose pD(A_R S))$ avec $A_S, A_R \in AGT$.

Description : Si l'étape est intéressante pour résoudre le problème, A_R l'ajoute à ses croyances.

Sémantique : Absent de Q_{ini} et Q_{int} .

- **query**

Syntaxe : $pMessage(A_S A_R query pD(A_S S))$ avec $A_S, A_R \in AGT$ et $S \in STA$.

Description : A_R ajoute à ses croyances la croyance que A_S désire une information à propos d'une étape.

Sémantique :

$$\frac{Q_{ini}(ini, s2, s4, s7) \xrightarrow{!query} Q_{ini}(s1)}{aUpdateTA(M)}$$

$$\frac{Q_{int}(ini) \xrightarrow{?query} Q_{int}(s1)}{aAdd(pB(pD(A_S, S))); aUpdateTA(M)}$$

$$\frac{Q_{int}(s1, s2, s4) \xrightarrow{?query} Q_{int}(s1)}{aAdd(pB(pD(A_S, S))); aUpdateTA(M)}$$

- **refine**

Syntaxe : $pMessage(A_S A_R refine pD(A_S S))$ avec $A_S, A_R \in AGT$ et $S \in STA$.

Description : A_R ajoute à ses croyances, la croyance qu'il a l'intention de raffiner une demande envoyée précédemment.

Sémantique :

$$\frac{Q_{int}(s1) \xrightarrow{!refine} Q_{int}(s4)}{aUpdateTA(M)}$$

$$\frac{Q_{ini}(s1) \xrightarrow{?refine} Q_{ini}(s7)}{aAdd(pB(pI(aRefine(S)))); aUpdateTA(M)}$$

- **refuseProposal**

Syntaxe : $pMessage(A_S A_R refuseProposal pD(A_S S))$ avec $A_S, A_R \in AGT$ et $S \in STA$.

Description : A_R supprime de ses désirs le désir qui a provoqué l'envoi du *propose*.

Sémantique : Absent de Q_{ini} et Q_{int} .

- **reply**

Syntaxe : $pMessage(A_S A_R reply pB(S))$ avec $A_S, A_R \in AGT$ et $S \in STA$.

Description : A_R ajoute à ses croyances l'étape contenue dans le message.

Sémantique :

$$\frac{Q_{int}(s1, s2) \xrightarrow{!reply} Q_{int}(s2)}{aUpdateTA(M)}$$

- **thank**

Syntaxe : $pMessage(A_S A_R thank pB(S|\neg S))$ avec $A_S, A_R \in AGT$ et $S \in STA$.

Description : Pas d'action particulière à réaliser à l'envoi/réception d'un *thank*. L'automate concerné sera fermé dans $aUpdateTA(M)$.

Sémantique :

$$\frac{Q_{ini}(s2) \xrightarrow{!thank} Q_{ini}(s3)}{aUpdateTA(M)}$$

$$\frac{Q_{int}(s2) \xrightarrow{?thank} Q_{int}(s3)}{aUpdateTA(M)}$$

Il existe toujours un état initial à partir duquel il est possible de recevoir la performative. Ainsi, à tout moment, la réception de n'importe qu'elle performative peut être traitée.

De plus, à la réception ou à l'envoi d'un message, les automates sont actualisés par l'action interne abstraite $aUpdate(M)$, $M \in MES$. Cette action vérifie si l'un des automates ouverts est dans un état à partir duquel l'agent peut envoyer ou recevoir le message. Si c'est le cas, l'automate est modifié en conséquence. Sinon un nouvel automate est ouvert.

5.4 Discussion

Ce modèle gère l'interaction humaine à la fois au niveau des énoncés et au niveau du discours. Au niveau de l'énoncé, un message est décrit par une performative appliquée à un état mental dont le type est lié au type de la performative. Au niveau du discours, les interventions sont représentées sous la forme d'automates temporisés. Ces deux niveaux sont liés par la sémantique des performatives.

6 Travaux Existants

Ces travaux sont à l'intersection de trois grands domaines : la modélisation cognitive, la planification coopérative humaine et l'interaction humaine. A notre connaissance, aucun travail n'intègre ces trois aspects en même temps.

Allen, Blaylock et Ferguson présentent dans [1] un modèle d'agent collaboratif intégrant un noyau de résolution individuelle de problèmes, un modèle de résolution collaborative de problème et un modèle d'interaction en langage naturel. Même si ce modèle ne décrit pas explicitement de modèle BDI, les notions utilisées (*situations*, *objectifs* et *recettes*) sont semblables.

Le principal défaut de ce modèle est qu'il est basé sur l'intuition qu'ont les auteurs du déroulement de la planification et des interactions humaines et non sur l'analyse des résultats d'une expérimentation réelle.

Le point de vue adopté par Bouzouba et Moulin dans [4] est proche du notre mais ils se focalisent uniquement sur un modèle de communication. Ils proposent d'étendre KQML à KQML+ pour répondre mieux à la théorie des actes de langage. Ils s'inspirent de conversations humaines pour concevoir leurs performatives mais sans utiliser de corpus réel.

De nombreux formalismes permettent de représenter une conversation. KQML [12] et FIPA-ACL [10] proposent des protocoles de communication qui ne prennent pas en compte le caractère dynamique des dialogues humains. Les Dooley Graphs introduits par Van Dyke Parunak [14] contiennent des informations sur la situation courante et les protagonistes d'une conversation. Cependant, ils ne tiennent pas compte de la dimension temporelle importante dans la dynamique conversationnelle. Notre modélisation par automates temporisés est motivée par la nécessité de représenter les durées associées aux actions communicatives et de synchroniser les interactions au sein des interventions.

Divers travaux concernent la sémantique des performatives. La théorie de Cohen et Levesque [7] a été une étape importante dans la formalisation des actes de langage. Cependant, comme le souligne Singh [19], leur théorie est conceptuellement problématique. La sémantique proposée par Singh [18] est orientée par la satisfaction des actes de langage. La sémantique de Dignum and Weigand [8] intègre logique illocutoire et logique déontique. La sémantique de Sadek [16] qui est celle utilisées par FIPA-ACL, contient des opérateurs modaux pour manipuler les croyances, désirs et incertitudes des agents.

7 Conclusion et Perspectives

Le modèle présenté dans cet article est fondé sur l'analyse de protocoles expérimentaux issus d'une expérimentation psychologique. Il suit les interactions humaines observées dans ces protocoles aussi fidèlement que possible. Les automates temporisés sont utilisés comme formalisme pour introduire la récursivité et la gestion du temps dans la représentation des conversations. Les performatives et les automates tem-

porisés sont liés par une sémantique donnée en termes de croyances, désirs et intentions.

Ce modèle est intégré dans l'architecture BDIGGY. Dans cette architecture, communication et planification sont représentées de manière homogène (*i.e* BDI). De plus, BDIGGY améliore l'architecture BDI classique en incluant un module de communication pour générer et interpréter des messages, et permet une construction dynamique des plans.

Nous travaillons actuellement à valider le modèle cognitif en comparant les protocoles expérimentaux et les protocoles générés artificiellement par la simulation.⁴

De plus, nous comptons vérifier que ce modèle n'est pas dépendant du problème lié à l'expérimentation mais peut être étendu à d'autres dialogues de recherche d'information.

Références

- [1] J. Allen, N. Blaylock, and G. Ferguson. A problem solving model for collaborative agents. In *AAMAS'02*, volume 2, pages 774–781, Bologna, Italie, 2002.
- [2] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical computer science*, 126 :183–235, 1994.
- [3] J. Austin. *Quand dire, c'est faire*. Seuil, Paris, 1970.
- [4] K. Bouzouba and B. Moulin. Les interactions distribuées : Approche basée sur le point de vue personnel des agents. In *MFI*, volume 1, pages 15–30, 2001.
- [5] M. Bratman, D. Israel, and M. Pollock. Plans and resource-bounded practical reasoning. *Computational intelligence*, 4(4) :349–355, 1988.
- [6] N. Chaignaud, I. Cheikhrouhou, and A. Nguyen-Xuan. Generalisation of a cognitive model for problem solving with incomplete information. In *ICCM'00*, pages 58–69, Groningen, Nederlands, 2000.
- [7] P. R. Cohen and H. J. Levesque. Performatives in a rationally based speech acts theory. In *Meeting of The Association for Computational Linguistic*, pages 79–88, Pittsburgh, Pensylvanie, 1990.

⁴Cette comparaison est faite au moyen d'un test "à la Turing" et d'un test statistique.

- [8] F. Dignum and H. Weiggand. Modelling communication between cooperative systems. In *Conference on Advanced Information Systems Engineering (CAISE'95)*, pages 140–153, 1995.
- [9] M. d’Inverno, D. Kinny, M. Luck, and M. Wooldridge. A formal specification of dmars. Technical note 72, Australian Artificial Intelligence Institute, Carlton, Victoria, 1997.
- [10] FIPA. Fipa specification : Agent communication language. Technical report, Foundation for Intelligent Physical Agents, 1999. Available at <http://www.fipa.org>.
- [11] A. Guerra-Hernandez, A. El Fallah Seghrouchni, and H. Soldano. Bdi multi-agent learning based on first-order induction of logical decision trees. In *IAT'01, IEEE Press*, pages 160–169, Maebashi, Japan, 2001.
- [12] Y. Labrou and T. Finin. A proposal for a new kqml specification. Technical report cs-97-03, UMBC, Baltimore, Maryland, 1997. Available at <http://www.cs.umbc.edu/kqml/papers>.
- [13] J. Moeschler and A. Reboul. *Dictionnaire encyclopédique de pragmatique*, chapter 1 : Théorie des langage, pages 43–78. Seuil, 1994.
- [14] H. V. D. Parunak. Visualizing agent conversations : using enhanced dooley graphs for agent design and analysis. In *Proceedings of ICMAS'96*, pages 275–282, Kyoto, Japon, 1996.
- [15] A. S. Rao and M. P. Georgeff. Modelling rational agents within a bdi architecture. In *Proceedings of KR'91*, pages 473–484, Cambridge (MA), 1991.
- [16] M. Sadek. A study in the logic of intention. In *KR'92*, pages 462–473. Cambridge, 1992.
- [17] J. Searle. *Les actes de langage*. Hermann, Paris, 1972.
- [18] M. P. Singh. Towards a formal theory of communication for multiagent systems. In *International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 69–74, Sidney, Australie, 1991.
- [19] M. P. Singh. A critical examination of the cohen-levesque theory of intention. In *European Conference on Artificial Intelligence (ECAI'92)*, 1992.
- [20] D. Vanderveken. *Analyse et simulation de conversations. De la théorie des discours aux systèmes multiagents*, chapter 2 : La structure logique des dialogues intelligents, pages 61–100. L’interdisciplinaire, Limonest, 1999.