Agents conversationnels psychologiques

Un cadre d'étude des comportements rationnels et psychologiques des agents assistants conversationnels

François Bouchet* — **Jean-Paul Sansonnet****

* *SMART Laboratory, Université McGill 3700 McTavish, Montréal, QC H3A 1Y2 Canada* francois.bouchet@mcgill.ca
** *Laboratoire LIMSI-CNRS BP 133 F-91403 Orsay Cedex* jps@limsi.fr

RÉSUMÉ. *Afin de faciliter l'accès et l'utilisation par les personnes du grand public des applications et services en informatique qui se répandent rapidement en particulier sur l'internet, il est nécessaire de proposer de nouveaux outils d'assistance qui offrent une interaction naturelle afin d'être mieux acceptés. L'approche des agents conversationnels semble prometteuse mais les agents ne peuvent pas se contenter d'opérer un raisonnement de type rationnel sur la structure et le fonctionnement des applications assistées. Ils doivent aussi exprimer des comportements psychologiques incluant des relations sociales, des traits de personnalité, des affects. Dans la première partie de l'article, nous proposons un cadre flexible pour modéliser les relations entre les réactions rationnelles et comportementales d'un agent assistant. Ensuite ce cadre est utilisé pour implémenter une première étude de cas, fondée sur la notion de biais cognitif.*

ABSTRACT. *In order to facilitate the access and the usage to the general public of the rapidly expanding applications and services, particularly on the internet, new assisting tools are needed with two main requirements: naturalness and acceptability. Conversational agents are a promising approach but assisting agents cannot rely only on rational reasoning over the structure and the functioning of the assisted applications. They must also express behavioral reactions that involve social relationships, character traits and affects. In the first part of this paper, we propose a flexible framework for modeling the relationships between the rational and behavioral reactions of an assisting agent. Then this framework is used to support a first case-study, based on cognitive biases.*

MOTS-CLÉS : *agents conversationnels, raisonnement rationnel et psychologique, biais cognitifs.*

KEYWORDS: *conversational agents, psychological and rational reasoning, cognitive biases.*

RSTI - RIA – 25/2011. Systèmes multi-agents : défis sociétaux, pages 591 à 623

1. Introduction

1.1. *Agents assistants conversationnels*

Ce travail est fondé sur la notion d'agent assistant conversationnel (AAC) qui se situe à l'intersection de deux domaines de recherche :

Agents assistants : il s'agit d'outils logiciels destinés à assister, de multiples manières, les gens ayant affaire à des activités informatiques ou médiées par de l'informatique (Maes, 1994). Les agents assistants se fondent sur les principes de raisonnement, tels que définis en intelligence artificielle, et se focalisent sur la notion d'agent rationnel (Bratman, 1987). En effet, en raison de la croissance actuelle des applications et des services informatiques et de leurs fonctionnalités qui s'accompagne aussi de l'élargissement de la population des utilisateurs du grand public, le besoin d'une assistance spécialement destinée aux novices, certes étudiée depuis longtemps (Cohill *et al.*, 1985), devient maintenant crucial.

Agents conversationnels : dans la communauté de recherche portant sur les agents conversationnels, dénotés en anglais « *Embodied Conversational Agents* » (ECA) (Cassell *et al.*, 2000), le terme conversationnel a pour sens particulier de désigner des outils logiciels en interaction avec des humains, en opposition par exemple avec les agents logiciels des systèmes multi-agents (SMA) qui interagissent essentiellement entre eux. Ici, il s'agit en fait d'étudier des personnages graphiques virtuels qui interagissent avec les gens lors de sessions impliquant plusieurs modalités : langue naturelle écrite ou parlée, gestes, émotions faciales, actions sur l'interface et dans l'environnement, etc. La recherche sur les agents conversationnels se focalise actuellement sur la modélisation de la psychologie humaine (états mentaux, émotions, etc.) (Ekman *et al.*, 1972; Frijda *et al.*, 1986) ainsi que sur son expression sous forme de personnages graphiques animés interagissant dans des sessions dialogiques qui font intervenir une large classe de situations interactionnelles.

Dans cette étude, on restreindra la notion d'assistance aux situations où l'usager est novice (*i.e.* une personne du grand public) et où les outils d'assistance sont explicitement appelés par l'usager[1]. Nous utilisons alors le mot *assistant* comme un terme générique qui englobe plusieurs sous-cas de situations interactionnelles comme par exemple :

– *agents présentateurs* : outils logiciels non interactifs affichant du contenu informationnel à propos d'un sujet (appelé par la suite le « topic ») ;

– *agents d'aide* : outils logiciels apportant une aide active aux usagers ayant besoin de remplir une tâche dans un environnement informatique qu'ils connaissent mal ;

– *agents majordomes* : des agents jouant le rôle d'assistants personnels, tels que les « *butlers* » ;

1. On écarte les outils d'assistance enfouis ou semi-intrusifs, comme les systèmes de recommandation dynamiques.

– *agents amis* : des agents logiciels jouant le rôle d'amis dans les activités ludo-sociales (*e.g. chatbots*, réseaux sociaux, etc.) ;

– *agents compagnons* : des partenaires (ou adversaires) dans les jeux de rôles, les jeux sérieux ;

– *agents tuteurs* : des agents jouant le rôle d'enseignants dans le cadre d'activités didactiques ;

– *agents entraîneurs* : des instructeurs dans les activités d'entraînement, en simulation par exemple ;

– *agents coachs* : mesurant et contrôlant la vie privée des gens (*e.g.* afin de réaliser un régime).

Toutes ces situations partagent un même schéma interactionnel, dénoté UAS : trois acteurs (Usager, Agent, Système) sont en interaction bilatérale via trois interfaces : graphique, dialogique, contrôle. La liste ci-dessus est longue mais a le mérite de montrer le spectre des modes interactionnels potentiels entre humains et agents, qui va de la situation où l'usager a un contrôle total sur l'assistant jusqu'à la situation inverse.

Dans ces situations interactionnelles, la modalité langagière continue à jouer un rôle important, même si les autres modalités sont de plus en plus mises en avant dans les recherches sur les agents conversationnels. De plus, si on les considère en toute généralité, la capacité de l'agent à supporter des sessions complexes de dialogue en langue naturelle est indispensable à terme. Cependant, dans cette étude, nous avons restreint notre discussion à des interactions langagières isolées (*i.e.* de type Usager : requête $\Rightarrow$ Agent : réaction), ceci pour deux raisons principales :

– la recherche sur la gestion du dialogue en langue naturelle est un domaine en soi qui développe des travaux sur la théorie de la conversation (Ogborn *et al.*, 1984) ou de l'analyse conversationnelle (Goodwin *et al.*, 1990), et sur les stratégies de la pragmatique du dialogue (*cf.* par exemple (Grau *et al.*, 1994; Caelen *et al.*, 2007) pour le français). Les travaux présentés ici mettent en avant l'idée qu'il existe une influence exercée, à divers degrés, par les notions psychologiques sur l'ensemble des comportements des agents. Ceci s'applique à tous les niveaux : résolution de requêtes formelles, planification d'actions sur l'environnement, expressions gestuelles, interactions langagières isolées ; bien sûr, elles ont aussi une influence sur la prise en compte de la pragmatique du dialogue. C'est donc l'étendue même du domaine concerné qui fait que nous avons restreint cette étude à des interactions langagières isolées.

– Un second argument est d'ordre pratique : dans des travaux précédents, portant sur les agents assistants (Bouchet, 2009), il est apparu dans le corpus de données issu des expérimentations que les usagers ont en fait très peu recourt à des dialogues complexes voire simples, la plupart des sessions étant limitées à des interactions langagières isolées parfois à deux tours[2]. Bien sûr, ceci ne se généralise pas à toutes les

2. En fait, dans les deux architectures d'agents assistants que nous avons développées InterViews (Sansonnet, 1999) et DAFT (Sansonnet *et al.*, 2006), nous avons implémenté une gestion simple du dialogue (fondée sur la notion de focus jointe à un traitement de l'anaphore à la

situations interactionnelles citées ci-dessus (par exemple, aux agents *amis* ou *compagnons*) mais c'est un indice qui indique que la restriction à des interactions langagières isolées n'est pas trop handicapante en pratique, même si elle se doit d'être abordée à terme.

1.2. *Agents rationnels et psychologiques*

La première qualité d'un agent assistant est de répondre de manière *efficace* aux sollicitations des usagers qui ont besoin d'aide. Pour ce faire, un agent assistant doit satisfaire deux besoins fondamentaux :

– il doit être muni de capacités de raisonnement symbolique sur la structure ainsi que le fonctionnement du système dont il est chargé d'assurer la fonction d'assistance auprès des usagers (Sansonnet, 1999). C'est la raison pour laquelle les architectures d'agents assistants sont fondées sur des modèles issus des recherches traditionnelles en intelligence artificielle, comme les systèmes à base de règles.

– Il doit aussi disposer de moyens d'interaction sophistiqués, dans la mesure où il interagit non pas avec d'autres agents logiciels mais avec des usagers, qui sont souvent des usagers novices en informatique ou tout au moins vis-à-vis du système qu'ils utilisent. Dans ce cas, l'impact positif de l'interaction en langue naturelle pour exprimer des requêtes d'assistance a été montré par Carbonell (2003) et pousse à aller vers une interaction de plus en plus intuitive. La conséquence est que les architectures d'agents assistants s'orientent de plus en plus vers les modèles d'agents implémentant les principes de *practical reasoning* de Bratman (Bratman *et al.*, 1988) ainsi que les modèles qui ont suivi comme les architectures de type BDI (Rao *et al.*, 1991). Dans la suite, nous référerons ces modèles par le terme d'**agents rationnels**.

La seconde qualité d'un agent assistant est de répondre de manière *pertinente* aux sollicitations des usagers qui ont besoin d'aide. Ce besoin est apparu lorsqu'on a commencé à se rendre compte que les utilisateurs se détournaient systématiquement des systèmes d'aides variés qui leur étaient proposés. Il a été montré que lorsque les utilisateurs novices ont besoin d'assistance, ils ont tendance à préférer s'adresser à « un ami derrière l'épaule », selon la terminologie de (Capobianco *et al.*, 2001) plutôt que d'avoir recours au système d'aide de leur ordinateur. Cette désaffectation met en avant le facteur d'*acceptabilité*, tel que défini par exemple par Davis (1989), considéré aujourd'hui comme un frein au développement des nouveaux outils d'assistance. En fait, il s'agit d'un phénomène qui s'inscrit dans un contexte plus large qui est celui de l'accès des personnes du grand public à une information de plus en plus foisonnante, par exemple via l'internet. Par leur foisonnement même, les entités informatiques produites entrent en compétition entre elles :

Hobbs (1978)) qui permet de résoudre la plupart des ellipses qui occurrent fréquemment lors d'un second tour de parole. Cependant cela ne peut servir de support à une étude de l'impact des aspects psychologiques sur la gestion du dialogue qui mérite une étude en soi.

– pour toucher leur public : c'est le problème de l'*accroche* (Galon, 1999) ;

– pour ne pas être délaissés tout de suite (*discarded*) : c'est le problème de l'*acceptabilité ergonomique* (Wu *et al.*, 2009) ;

– pour être pris au sérieux : c'est le problème de la *crédibilité* (Fogg *et al.*, 2003) ;

– pour qu'ils soient correctement assimilés : c'est le problème du *but communicatif* (Coiro, 2003).

On se trouve alors devant un problème qui devance même la question de l'efficacité de l'agent puisqu'il s'agit avant tout qu'il ne soit pas rejeté prématurément, à la manière du Clippie-effect (Galluccio, 2006). Dans la suite, nous référerons cette question par le terme « interaction pertinente ».

Les agents conversationnels (dénotés plus haut ECA) constituent un domaine de recherche très actif (Cassell *et al.*, 1999). Les recherches ont pu établir qu'ils possèdent de bonnes propriétés vis-à-vis de la question de l'interaction pertinente, par exemple, concernant : l'intérêt (*user-friendliness*) (Lester *et al.*, 1997) utile pour l'accroche, l'acceptabilité (Davis, 1989), la crédibilité (Hayes-Roth, 2004), et enfin la performance accrue du couple humain-agent (Buisine *et al.*, 2005), utile pour l'assimilation de notions. À partir de là, il devient envisageable d'utiliser des ECA afin d'offrir une assistance plus naturelle aux usagers novices, ce qui a conduit à définir une sous-classe des ECA, dédiée à la fonction d'assistance : les agents assistants conversationnels (AAC).

De manière similaire aux études qui ont mis l'accent sur l'amélioration de la crédibilité (*believability*) physique des ECA, par exemple via l'expression d'émotions (Bates, 1994; Martin *et al.*, 2007), nous pensons qu'il faut développer maintenant des agents qui soient non seulement physiquement crédibles, mais aussi cognitivement crédibles, c'est-à-dire capables d'exhiber des comportements complexes, si possible similaires à ceux des humains. Pour cela, il faut les doter d'une personnalité psychologique (John *et al.*, 2008) qui les rendent capables d'interagir avec les usagers en accord avec le *personnage* qu'ils endossent : rôle social, traits de personnalité, affects (*i.e.* les sentiments et les émotions), etc. Dans la suite, nous référerons ces agents par le terme d'**agents psychologiques**.

Le plan général de l'article est le suivant : la section 2 est consacrée à l'état de l'art sur les recherches qui visent à doter des agents rationnels de capacités psychologiques au sens large du terme. La section 3 présente une architecture dédiée à l'implémentation et à l'expérimentation d'agents rationnels et psychologiques (le *framework R&B*). La section 4 décrit le modèle d'agent *R&B*. Dans la section 5, l'approche est mise à l'épreuve via l'implémentation d'un scénario *R&B* particulier qui est fondé sur la notion de biais cognitif. Enfin, dans la section 6, des éléments concernant l'implémentation ainsi que les travaux connexes sont discutés.

2. Études sur les agents rationnels et psychologiques

2.1. *Architectures d'agents rationnels*

Dans notre contexte, le terme « agent rationnel » correspond aux définitions classiques en intelligence artificielle (IA) et systèmes multi-agents (SMA) du *raisonnement pratique* de Bratman déjà cité. Par exemple, la définition de Russell et Norvig (Russell *et al.*, 2009) (pp. 38-39) ou celle de Wooldridge qui adapte le raisonnement pratique aux SMA : « Un agent est dit rationnel s'il choisit les actions qui satisfont son propre intérêt, étant donné les croyances qu'il a sur l'état courant du monde » (Wooldridge, 2000) (p.1) et il propose un modèle formel (Wooldridge *et al.*, 1995), fondé sur la théorie BDI de Rao et Georgeff (Rao *et al.*, 1995). En fait, le *raisonnement pratique* est souvent opposé au *raisonnement théorique* qui vise la consistance des représentations du monde et qui n'est pas discuté ici. De même, nous n'aborderons pas la notion d'agent rationnel telle qu'elle s'entend dans la théorie de la décision, où un agent maximise une fonction d'utilité. Ces deux derniers types de raisonnement rationnel nécessitent en effet une connaissance globale et totale du monde (cependant la théorie de la décision est beaucoup utilisée dans les études SMA sur les macromodèles de socio-économie).

Les architectures d'agents rationnels sont souvent fondées sur SOAR (*State, Operator And Result*) de Laird qui fut le premier modèle populaire destiné à la modélisation des processus cognitifs au moyen de règles (Laird *et al.*, 1987). ACT-R (*Adaptive Control of Thought-Rational*) d'Anderson (1996) fonctionne sur les mêmes principes, avec une séparation claire entre les représentations déclaratives et procédurales.
Faisant suite à SOAR et ACT-R, la théorie BDI (pour *belief*, *desire* et *intention*) de Bratman, Rao et Georgeff (cités ci-dessus) met en avant la primauté des désirs et des intentions dans l'action rationnelle. Les plateformes BDI sont nombreuses et utilisent des logiques (KGP, MINERVA, AGENTSPEAK, CAN ; avec des règles : 2APL ; ou temporelle : MetateM, la famille Golog) ou simplement Java (JASON, JADEX, JADE, JACK). Toutes ces architectures sont fondées sur le cycle de délibération BDI initial.

2.2. *Introduction de capacités psychologiques dans des agents rationnels*

Dans leurs travaux, Gratch et Marsella (Gratch *et al.*, 2004) ont proposé un modèle d'émotions fondé sur l'architecture SOAR. Cela a débouché sur d'importantes difficultés au niveau de l'architecture interne, soulignées par Laird lui-même dans (Gratch *et al.*, 2005). Cela a donc favorisé l'utilisation des architectures de type BDI, en particulier sous l'influence des chercheurs en SMA. En effet, ceux-ci ont exploré depuis le début la notion « d'agent cognitif » au moyen de théories destinées à modéliser les capacités de raisonnement des agents en s'appuyant généralement sur des architectures BDI. Remarquons que les termes *belief*, *desire* et *intention* ne sont pas réellement pris ici au sens psychologique ; ils correspondent de fait à des notions qui restent propres à l'action rationnelle (faits, buts, plan en cours).

Récemment, les chercheurs en SMA se sont intéressés à l'intégration de notions plus proches de la psychologie dans des architectures cognitives d'agents, par exemple, en ajoutant une couche à des plateformes SMA déjà existantes, comme dans le cas de CoJACK (Norling *et al.*, 2004; Evertsz *et al.*, 2008) pour la plateforme JACK qui prend en compte des paramètres émulant certaines caractéristiques de la physiologie humaine comme : la durée d'apprentissage ; les limites mémorielles (*e.g.* « perdre une croyance » si l'activation est basse ou encore « oublier l'action suivante » dans une procédure) ; la remémoration vague de croyances (*fuzzy retrieval*) ; la limitation du champ d'attention ; ou encore l'utilisation d'opérateurs, appelés modérateurs, qui altèrent le raisonnement cognitif lui-même.

Des études ont aussi été menées pour ajouter des émotions aux architectures d'agents BDI, par exemple pour prendre en compte des notions comme : la peur, l'anxiété ou encore la confiance en soi. Ceci a été effectué par l'ajout de paramètres supplémentaires comme les désirs fondamentaux, les capacités et les ressources. Une autre voie (Casali *et al.*, 2004) a consisté à ajouter des degrés aux logiques multivaluées représentant les croyances, désirs et intentions, en particulier en s'appuyant sur la logique de Lukasiewicz (1963).

De manière indépendante, un certain nombre d'études ont porté sur l'introduction de capacités psychologiques dans des agents conversationnels. Dès le milieu des années 1990, Maes (1994), Bates (1994) puis Hayes-Roth (Hayes-Roth *et al.*, 1995) et Rousseau (Rousseau *et al.*, 1996) ont introduit la notion de *believable agents*, qui pose le principe que les traits de personnalité ont une influence non négligeable sur la prise de décision dans l'action rationnelle (sans aller cependant jusqu'aux positions extrêmes de Damasio (1994)).

Très récemment, les travaux de Pélachaud *et al.* (McRorie *et al.*, 2009; Bevacqua *et al.*, 2010), fondés sur l'architecture d'agent conversationnel GRETA (Pelachaud, 2000), font intervenir des modèles de personnalité pour l'expression des émotions (faciales, gestuelles etc.). S'appuyant dessus, avec l'architecture FATIMA, Paiva *et al.* (Doce *et al.*, 2010) affirment qu'il est indispensable de construire des agents qui ont des apparences physiques bien distinctes mais aussi des comportements psychologiques bien différenciés : « *in the era of globalization, concepts such as individualization and personalization become more and more important in virtual systems. The FFM* [*cf.* section 4.2] *can be a basis for the creation of distinguishable personalities by using the personality traits to automatically influence cognitive processes : appraisal, planning, coping and bodily expression* ».

2.3. *Relations entre rationalité et psychologie dans un contexte dialogique*

Reprenant à notre compte la citation de Paiva *et al.* ci-dessus, nous la replaçons dans le contexte particulier des agents assistants conversationnels où la modalité dialogique (conversation en langue naturelle) joue un rôle essentiel. Ceci nous a amené à développer un outil dédié aux expérimentations sur les agents conversationnels, dans

le cadre de l'internet. Un toolkit appelé DIVA (Sansonnet, 2008), a été développé avec deux objectifs principaux :

1) *il est orienté-web*, c'est-à-dire dédié au développement d'AAC pour des applications et services dans l'internet. L'agent assistant est alors personnifié par un personnage virtuel animé à l'écran qui est totalement intégré à la page web. L'agent peut interagir en langue naturelle avec les usagers (ils tapent leurs requêtes dans un champ texte – ou *chatbox* – et l'agent répond textuellement aussi grâce à un phylactère).

2) *Il est open-source*, c'est-à-dire disponible librement, pour les besoins de la recherche et de l'expérimentation sur les AAC.

L'outil DIVA s'est avéré simple et efficace d'emploi ce qui a permis de développer plusieurs applications expérimentales (*cf.* exemples sur le site web de DIVA (Sansonnet, 2008)). Cependant les premières expérimentations avec des usagers novices (Xuetao *et al.*, 2009) ont fait apparaître certaines limitations des agents assistants lorsqu'ils sont fondés sur un moteur strictement rationnel. Elles ont mis en lumière au moins trois phénomènes saillants qui gênent la perception de « l'humanité » (*human-likeness*) et de la naturalité dialogique des agents :

1) *répétition des réactions coopératives de l'agent :* l'agent est toujours servile (*responsive*) quoi qu'il puisse se passer pendant la session, quel que soit le rôle qu'il endosse et ceci quoi que lui demande ou dise l'usager (par exemple, l'agent ne peut cacher une information qu'il possède à l'usager ni s'empêcher d'exécuter une action qu'il peut faire si l'usager le lui demande) ;

2) *répétition des schémas de réponses :* l'agent fournit la même information plusieurs fois de suite sans aucune variation linguistique ;

3) *répétition des réactions rationnelles :* si l'usager pose plusieurs fois de suite la même requête, à chaque fois l'agent réagit à la requête indépendamment de la session dialogique, c'est-à-dire comme si la requête n'avait jamais été posée auparavant.

Ces observations montrent l'importance que le grand public apporte à la perception de comportements jugés non naturels lors de leurs interactions dialogique avec des agents conversationnels chargés de les assister. Dans la section suivante, nous proposons une architecture d'agent assistant psychologique qui tente d'intégrer à la fois le comportement rationnel, indispensable, et le comportement psychologique.

3. Architecture d'agent assistant psychologique

3.1. *Architectures pour le dialogue homme/machine*

Dans la section 1.1, nous avons précisé les limites du domaine de traitement du dialogue homme/machine pour cette étude qui se restreignent à la prise en compte de requêtes textuelles isolées. Le champ des architectures pour le dialogue homme/machine textuel a été considérablement développé depuis les vingt dernières années et ont produit des systèmes opérationnels ; nous en citerons trois :

– TRIPS/TRAINS de Allen et Ferguson (Allen *et al.*, 1995; Ferguson *et al.*, 1998). Les projets TRIPS et TRAINS visent à développer des systèmes de dialogue génériques, c'est-à-dire qui ne dépendent pas d'une application particulière : pour ce faire, les auteurs ont essayé de décomposer les phases de traitement de la langue en divers modules dont certains peuvent être réutilisés d'une application à l'autre et d'autres doivent être redéfinis pour chaque nouvelle application ; ces modules sont consacrés au traitement syntactico-sémantique mais aussi au traitement des références, à la résolution de tâches et au raisonnement pragmatique en situation de dialogue finalisé (où l'utilisateur a une tâche à accomplir), etc. Les auteurs se focalisent sur la notion de compréhension profonde de la langue qui débouche sur une représentation sémantique formelle des phrases analysées qui est particulièrement précise. Ce système a été appliqué avec succès pour des tâches complexes de planification, en particulier dans le domaine ferroviaire, de même qu'à la gestion de ressources en situation de crise. Le projet TRAINS constitue un bon exemple de réalisation complexe et aboutie.

– COLLAGEN de Rich et Sidner (Rich *et al.*, 1997; Rich *et al.*, 2001), est un système de dialogue Homme/Machine finalisé qui repose résolument sur le concept d'agent. Dans COLLAGEN, le système est vu comme un agent assistant qui essaye de réaliser des tâches pour l'usager en s'appuyant sur une gestion du dialogue complexe qui fait intervenir des théories conversationnelles, comme la *Collaborative Discourse Theory* développée précédemment par Sidner et Grosz (Grosz *et al.*, 1986). Le principe de base de COLLAGEN, appelé *Collaborative Interface Agent* est similaire au schéma UAS défini à la section 1.1. Il a permis aux auteurs de développer de nombreuses applications d'agents conversationnels grâce à un système souple, fondé sur les Java Beans.

– TRINDI de Larsson et Traum (Larsson *et al.*, 2001), puis SIRIDUS (Quesada *et al.*, 2000), a été développé dans le cadre d'un projet de recherche européen et a débouché sur un système opérationnel, dont le logiciel est accessible librement sur internet[3] (le noyau de TRINDI est développé en Prolog). Les auteurs du projet ont essayé de construire une architecture générique, c'est-à-dire suffisamment flexible pour pouvoir implémenter différentes politiques de dialogue et les expérimenter. TRINDI est fondé sur trois notions : a) un modèle de l'état de la session dialogique (*information state*) ; b) un ensemble de règles qui consultent et modifient le modèle (*update rules*) ; c) un module permettant de superviser la politique globale de dialogue (*control algorithms*). Plusieurs études se sont appuyées sur ce logiciel pour développer des théories linguistiques spécifiques comme GoDiS (Larsson *et al.*, 1999), fondée sur la notion de sujet-en-cours de Ginzburg ou QUD (Cooper *et al.*, 2000).

Les systèmes de dialogue homme/machine textuel présentés ci-dessus présentent des caractéristiques très similaires au niveau des objectifs (généricité, opérationnalité) et de l'architecture globale (modularité, modèle de tâche, redéfinition possible des politiques de dialogue etc.). Alors que TRAINS reste la référence historique, COLLAGEN a introduit la notion d'agent assistant et TRINDI propose un logiciel configurable et accessible sur internet. Dans le cadre de notre étude, il n'est pas possible de

3. http ://www.ling.gu.se/projekt/trindi/trindikit/documentation.html

prendre en compte immédiatement toute la complexité de tels systèmes lorsque nous proposons des pistes qui permettent d'intégrer l'influence des notions psychologiques. C'est la raison pour laquelle, dans la section 3.2, nous décrivons une architecture qui limite les notions utilisées dans la suite de l'exposé et dans la section 3.3, nous décrivons le langage formel de requêtes où l'accent est mis sur le traitement de requêtes textuelles isolées.

3.2. *Agents R&B*

Dans cette section nous présentons une architecture, appelée framework *R&B*, qui constitue un cadre d'étude et d'expérimentation des relations souvent très intriquées (Frijda, 2006) (Scherer *et al.*, 2001) entre le raisonnement rationnel (au sens du *practical reasoning* de Bratman (Bratman *et al.*, 1988)) et le raisonnement psychologique. En s'appuyant sur une architecture d'agent rationnel proche du modèle standard BDI (Rao *et al.*, 1995), nous avons développé le framework *R&B* selon deux principes :

– *principe de séparation* : afin d'éviter les confusions entre concepts ressortant du domaine rationnel et du domaine psychologique ainsi que pour séparer les compétences des concepteurs (intelligence artificielle ou psychologie computationnelle), l'agent est composé de deux moteurs distincts (*rational engine* et *behavioral engine* – qui ont donné le nom au framework *R&B* pour « *Rational and Behavioral agents* ») qui exécutent chacun des heuristiques propres à un domaine ;

– *principe de généricité* : bien sûr, il s'agit ensuite d'étudier comment se font les interactions entre les deux moteurs. Pour cela, le framework *R&B* doit être capable de supporter l'élaboration de différentes stratégies, expérimentées pour évaluation.

La figure 1 présente une architecture typique d'AAC incorporé dans le framework *R&B*, composée de deux parties principales : l'interface usager et l'agent assistant.

L'interface usager ($\mathscr{I}$) : a pour fonction de gérer l'interaction entre le système d'aide proprement dit et l'utilisateur. Les agents conversationnels sont intrinsèquement multimodaux (Kopp *et al.*, 2004) mais dans le cas des agents assistants, la langue naturelle joue un rôle majeur dans le processus d'assistance parce que les usagers novices préfèrent exprimer leurs problèmes en langue naturelle, surtout en présence d'un ECA comme déjà souligné par Capobianco et Carbonell (Capobianco *et al.*, 2001). C'est pourquoi, pour simplifier, nous ne traitons pas les aspects multimodaux dans cet article mais seulement la modalité linguistique. Dès lors, les deux principaux modules de $\mathscr{I}$ sont :

– la chaîne de traitement de la langue naturelle (NLP-chaîne) : elle traduit les phrases tapées par l'usager en une structure formelle, appelée « *Formal Request Language* » (FRL), destinée à l'agent assistant ;

– la chaîne d'expression en langue naturelle (NLE-chaîne) : elle traduit les réponses formelles de l'agent assistant (représentées elles aussi en FRL) en faisant sur-

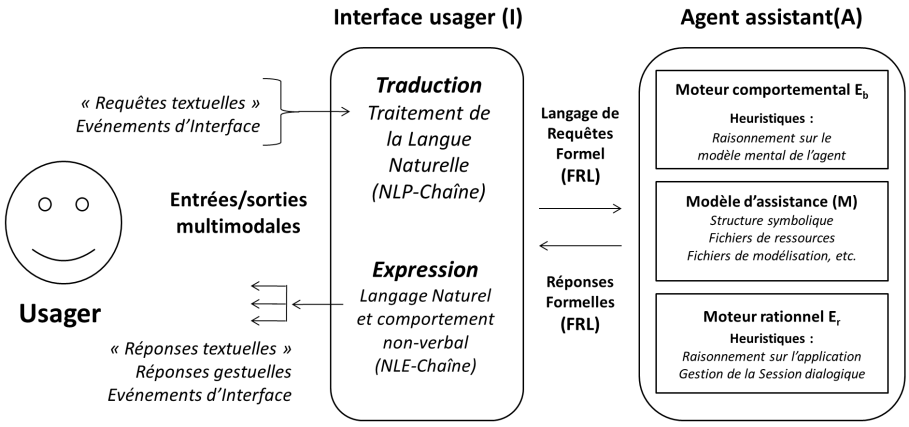


Figure 1. *Un agent assistant dans le framework R&B*

tout usage de la modalité linguistique (la NLE-chaîne n'est pas abordée ici).

L'agent assistant ($\mathscr{A}$) : est chargé de traiter les requêtes d'aide des usagers une fois qu'elles ont été traduites en FRL. L'assistant est lui aussi composé de trois parties principales :

– le modèle d'assistance ($\mathscr{M}$) : est un modèle formel de l'application, dédié au support de la fonction d'assistance. Si on considère plusieurs applications à assister, tous leurs modèles partagent la même ontologie, mais un modèle spécifique doit être instancié pour chacune des applications.

– Le moteur rationnel (*Rational Engine* $\mathscr{E}_r$) : traite les requêtes formelles au moyen d'heuristiques de raisonnement symbolique qui opèrent sur le modèle $\mathscr{M}$; ensuite il construit une réponse formelle qui est retournée à l'interface qui à son tour, exprime la réponse à l'usager.

– Le moteur comportemental (*Behavioral Engine* $\mathscr{E}_b$) : est composé d'heuristiques de raisonnement symbolique qui opèrent sur la partie du modèle $\mathscr{M}$ qui représente la psychologie (ou modèle mental) de l'agent ; les heuristiques comportementales influencent l'exécution rationnelle.

3.3. *Langage formel de requêtes*

Comme illustré à la figure 1, le langage formel de requêtes (noté FRL) constitue le canal d'entrée/sortie entre l'interface $\mathscr{I}$ et l'agent $\mathscr{A}$. Comme ce nous nous intéressons ici à l'architecture de l'agent, nous ne donnons que les notations utilisées dans les sections suivantes. Le langage FRL est le résultat de notre travail précédent portant sur la conception d'une NLP-chaîne pour des agents assistants conversationnels. Cette chaîne est fondée sur un corpus spécifique de 11 000 phrases de demande d'aide

Syntaxe	Glose [définie de manière similaire aux synsets Wordnet]	Catégorie
ask R|A|P	L demande une information ou vérifie si une proposition P est vraie ou si l'action A est connue de l'interlocuteur	
tell P	L affirme une information que la proposition P est vraie en ce moment	Connaissances
reply V	L donne une valeur V en réponse à une requête précédente (ask,…) de l'interlocuteur	
know R|A|P	L affirme qu'il a des informations au sujet du contenu (pour P, indique qu'il pense que P est vraie)	
unknown R|A|P	L affirme qu'il ne sait rien au sujet du contenu (ou au sujet du fait que P est vraie ou fausse)	
mistrust P|V	L affirme qu'il croit que la valeur V est erronée ou que la proposition P est probablement fausse	
why P|V	L demande pourquoi la proposition P est vraie ou pourquoi la valeur V a été donnée comme réponse	R référence A action P proposition V valeur
possible A	L demande si il est possible (disponibilité, droits, …) pour lui ou l'interlocuteur d'exécuter l'action A	
how A	L demande comment réaliser l'action/procédure A	
effect A	L demande quelles seraient les conséquences de l'exécution de l'action A	
execute A|R	L commande à l'interlocuteur d'exécuter l'action A ou d'activer la fonction principale de l'objet référé	Action
repeat -	L commande à l'interlocuteur d'exécuter la dernière action exécutée	
undo -	L commande à l'interlocuteur de défaire la dernière action exécutée	
suggest A|P	L encourage/suggère/permet à l'interlocuteur d'exécuter l'action A/d'adopter la proposition comme but	
object A|P	L décourage/interdit à l'interlocuteur d'exécuter l'action A/d'adopter la proposition P comme but	
intent A|P	L affirme qu'il a l'intention d'exécuter l'action A dans un futur proche/ qu'il a adopté P comme but	
judge P	L exprime une opinion subjective en affirmant la proposition P	Sentiments
feel P	L exprime un sentiment subjectif en affirmant la proposition P	
like R|A|P|V	L exprime une préférence/goût subjectifs a propos du contenu (sous cas de judge)	
dislike R|A|P|V	L exprime une non-préférence/dégoût a propos du contenu (sous cas de judge)	
bravo -	L congratule l'interlocuteur au sujet du topic (généralement le dernier sujet référé)	
criticize -	L critique l'interlocuteur au sujet du topic (e.g. contient les injures)	
agree -	L répond oui à une question oui/non ou est d'accord avec le topic	Dialogue
disagree -	L répond non à une question oui/non ou n'est pas d'accord avec le topic	
greet	L salut l'interlocuteur au début de la session	
bye	L demande à quitter la session	

'L' dénote le locuteur (i.e. l'usager ou l'agent dans les réponses)
'-' dénote le concept saillant dans la session dialogique (e.g."ça")

Figure 2. *Liste des performatifs du langage FRL et leurs gloses associées*

collectées d'une part à partir de plusieurs thésaurus et, d'autre part, d'expériences menées avec des usagers novices. Ce travail nous a permis d'effectuer une analyse et une classification du domaine linguistique concerné par la fonction d'assistance (Bouchet *et al.*, 2009) ce qui a débouché sur la définition du langage formel de requêtes. Ci-dessous nous décrivons seulement les notions de base de FRL, faisant abstraction des requêtes complexes comme le discours rapporté, les commandes conditionnelles, etc.

Une requête FRL de base est de la forme $F(X)$:
– F correspond à un performatif,
– X est le contenu du performatif.
Dans une requête FRL de base, il y a un seul argument.

Dans une requête, il y a quatre sortes de contenus possibles : $X \in \{R, A, P, V\}$
– R **référence** est une expression référentielle dans le modèle $\mathscr{M}$;
– A **action** est la description d'une action exécutable dans le système ;
– P **proposition** est une proposition logique portant sur l'état courant du modèle $\mathscr{M}$;
– V **valeur** est une valeur $\in V$ telle que définie dans le modèle $\mathscr{M}$;

Sujets des requêtes : l'émetteur de la requête (appelé le locuteur) est toujours le sujet du verbe associé au performatif : $F(X) \equiv F_{\text{CURRENTLOCUTOR}}(X)$. Selon le tour de dialogue, il peut s'agir de l'usager $\mathscr{U}$ ou bien de l'agent $\mathscr{A}$. Lors d'un tour de dialogue, plusieurs requêtes FRL peuvent être envoyées sous forme d'une séquence (séparées par un point-virgule).

Performatifs : leur liste (donnée au complet à la figure 2) est divisable en quatre grandes catégories (pouvant parfois se recouvrir) : savoir, action, affect et dialogue.

Quand l'usager tape une phrase dans la *chatbox* de l'interface graphique de l'agent, la NLP-chaîne l'associe à une requête FRL (ou une séquence en cas de besoin) puis des opérations de nature sémantique sont effectuées comme par exemple le traitement des indexicaux (*i.e.* remplissage de « je », « toi/vous », ...). Par exemple :
ENTREE USAGER « *Je t'aime bien* » $\Longrightarrow$ $\mathrm{LIKE_u[agent]}$
SORTIE AGENT $\mathrm{FEEL_a[M_h^{>}]} \Longrightarrow$ « *Je suis content* ! » (*cf.* notations section 5).

4. Modèle d'agent rationnel et psychologique

4.1. *Le modèle d'assistance*

Le modèle d'assistance ($\mathscr{M}$) de l'agent est une structure symbolique qui supporte, dans un cadre unifié, la représentation symbolique de toute l'information accessible à l'agent. Cette section est restreinte à l'exposé des notations nécessaires pour la suite. Nous allons présenter successivement :

– la syntaxe et le squelette de l'ontologie du modèle ;

– la dynamique du modèle, *i.e.* comment la représentation symbolique évolue ;

– les opérateurs de base du langage de questions du modèle (MQL pour *Model Query*[4] *Language*) ;

– les objets-questions, qui sont des formules MQL qui peuvent être manipulées, en tant que telles, par les deux moteurs : $\mathscr{E}_r$ et $\mathscr{E}_b$.

Plusieurs formalismes ont été proposés pour la représentation des connaissances : par exemple les logiques, les réseaux sémantiques, etc. Récemment, les structures arborescentes se sont répandues largement dans les applications internet afin d'augmenter la puissance de la représentation de base que constitue le DOM (RDF, OWL...). Dans la mesure où nous nous intéressons à l'assistance pour les applications web qui reposent sur le DOM et XML, nous avons choisi une forme arborescente pour l'implémentation du modèle. Cependant, ce modèle pourrait être aisément transposé vers d'autres formalismes de représentation des connaissances, comme par exemple une base de faits en Prolog.

Le modèle d'assistance $\mathscr{M}$ est un arbre dont :

– les nœuds non terminaux sont étiquetés par les concepts. Un concept est représenté par un symbole ou un entier (dans le cas où on a besoin d'indexer des listes de sous-concepts) ;

– les nœuds terminaux portent classiquement les valeurs des concepts : symboles, nombres, booléens, chaînes.

4. Dans la suite de l'exposé, nous traduirons le terme *query* par *question*, qu'il ne faut pas confondre alors avec des questions posées en langue naturelle par les usagers.

Nous utilisons une syntaxe à base de crochets pour représenter l'arbre (*e.g.* comme en Mathematica) :

```
Model = Rootconcept[
    Concept1[
        Concept11[...],
        Concept12[...],
    ...]
    Concept2[
        Concept21[...],
        Concept22[...]
    ...],
...]
```

Dans cette structure, chaque nœud est composé d'une tête (le symbole du concept) et d'un corps (le sous-arbre contenu entre les crochets), aussi appelé la valeur du concept. Ainsi, les valeurs de concepts peuvent être soit des valeurs terminales conventionnelles, soit des sous-arbres. Cette structure formelle permet d'organiser les concepts sous forme d'une ontologie statique qui constitue le modèle d'assistance effectif. Le premier niveau de l'ontologie est défini par le concept racine $\mathscr{M}$ et cinq sous-domaines ordonnés, considérés comme pertinents pour un AAC. Ainsi, le modèle est un quintuplet $\mathscr{M} =< \mathscr{A}, \mathscr{U}, \mathscr{R}, \mathscr{S}, \mathscr{T} >$ où :

1) *l'agent ($\mathscr{A}$)* contient l'information disponible au sujet de l'agent en tant que personnage : comme son nom, âge ou genre, mais aussi ses attributs comportementaux : traits de caractère, humeurs, etc.

2) *L'usager ($\mathscr{U}$)* contient l'information disponible au sujet de l'usager : comme son nom, ses préférences, etc.

3) *La requête ($\mathscr{R}$)* contient l'information disponible au sujet de la requête courante de l'usager : sa représentation textuelle et formelle mais aussi l'information concernée par son état courant de résolution.

4) *La session ($\mathscr{S}$)* contient l'information disponible au sujet des requêtes précédentes depuis le début de la session dialogique.

5) *Le topique ($\mathscr{T}$)* contient l'information spécifique disponible au sujet de l'application assistée.

Il convient de remarquer que le terme « disponible » employé ci-dessus met en exergue le fait que le modèle ne contient pas forcément l'information exhaustive au sujet des entités concernées mais seulement l'information qui a été fournie par les concepteurs de l'application ainsi que l'information que l'agent a été capable de récolter au cours de la session dialogique. C'est la raison pour laquelle l'agent doit raisonner en mode de connaissances incomplètes au sujet du monde (par exemple, on ne peut faire l'hypothèse CWA – *Closed World Assumption* (Reiter, 1978)).

Dynamique du modèle : le modèle $\mathscr{M}$ est une structure d'arbre évolutif, à la manière des algèbres évolutives (Gurevich, 1995). Étant donnée une nouvelle session, le modèle $\mathscr{M}_0$ démarre à t_0, tel que $\mathscr{M}_0 =< \mathscr{A}_0, \varnothing, \varnothing, \varnothing, \mathscr{T}_0 >$, où :

Nom	rôle	Valeur retournée
`GET[path]`	renvoie le sous-arbre à partir de n	`OK[expr]` | `FAIL[rapport]`
`SET[path, expr]`	remplace n par `expr`	`OK[expr]` | `FAIL[rapport]`
`MAP[path, func]`	remplace n par func(n)	`OK[func(n)]` | `FAIL[rapport]`
`ADD[path, expr]`	ajoute `expr` à n	`OK[expr]` | `FAIL[rapport]`
`DEL[path,` s_i`]`	supprime le sous-arbre ayant pour tête le symbole s_i dans n	`OK[expr]` | `FAIL[rapport]`
`VOID[]`	ne fait rien	toujours `OK[]`

Tableau 1. *Principales questions MQL.* `path` *correspond à une expression de chemin dans l'arbre* $\mathscr{M}.s_1.s_2,..,s_n$ *(*s_i *étant des symboles étiquetant les nœuds),* `n` *est le nœud référé par* `path` *et* `expr` *est une valeur terminale ou un sous-arbre*

– $\mathscr{A}_0$ est le sous-modèle générique de l'agent : il est défini par les concepteurs du système d'aide et ne dépend pas des applications assistées.

– $\mathscr{T}_0$ est le sous-modèle spécifique de l'application assistée créé par le programmeur et instancié avec les informations dédiées à l'assistance.

La structure arborescente du modèle évolue selon deux sortes d'évènements[5] :

1) les mises à jour effectuées par l'agent dans $\mathscr{A}$, $\mathscr{U}$, $\mathscr{R}$, $\mathscr{S}$ en fonction des interactions avec l'usager ;

2) les mises à jour effectuées sur les variables de $\mathscr{T}$ changées en fonction de l'évolution du *runtime* de l'application elle-même.

Le langage de questions du modèle : rappelons que le modèle est accédé par l'agent ou par l'application via une API appelée MQL et que, de même, une formule MQL sera appelée une *question* et non une *requête*, afin de ne pas la confondre avec une requête de l'usager, en particulier une requête FRL. Les principales questions de MQL sont résumées dans le tableau 1. Des questions supplémentaires comme `APPEND`, `FIRST`, `REST`, etc. facilitent le traitement des structures de listes (sous forme de nœuds énumérés). De même, les chemins d'accès dans les questions peuvent être abrégés lorsqu'il n'y a pas d'ambigüité : $\mathscr{M}.\mathscr{A}$.name $\rightarrow$ $\mathscr{A}$.name (réfère au nom de l'agent dans le modèle). De plus, parce que l'arbre est parcouru en mode « *depth-first* », « name » sera le nom de l'agent et non celui de l'usager.

Les objets-questions de l'agent : dans l'agent, les questions MQL sont générées par le processus rationnel ou bien par le processus comportemental. En fait, une fois

5. Ainsi, la structure du modèle est partagée entre l'application et l'agent qui fonctionnent comme deux processus séparés. La question de leur synchronisation est un problème en soi ; voir (Sansonnet, 2004) pour plus d'information à ce sujet. La synthèse du modèle d'assistance $\mathscr{M}_0$ à t_0 ainsi que sa mise à jour pendant la session, à $t_1,..,t_n$ définissant la séquence d'arbres $\mathscr{M}_1,..,\mathscr{M}_n$ est aussi une question en soi qui déborde du cadre de cet article (voir la thèse de Leray (Leray *et al.*, 2006)). C'est pourquoi dans la suite, nous référerons au modèle comme étant $\mathscr{M}$ alors que cela devrait être $\mathscr{M}_n$, l'arbre courant issu des n précédents évènements.

	Unaire	Binaire
Statique	Traits Ψ_T	Rôles Ψ_R
Dynamique	Humeurs Ψ_h	Affects Ψ_a

Tableau 2. *Les quatre types d'états mentaux*

créées, elles ne sont pas envoyées telles quelles au modèle pour exécution directe. Au lieu de cela, une question donnée Q_i est réifiée en un *objet-question* qui enveloppe la formule MQL dans une structure symbolique, exprimée dans le même formalisme que celui de $\mathscr{M}$, et qui fournit des attributs de gestion supplémentaires (par exemple, l'état d'exécution de la question, l'état résultant de l'exécution etc.). De cette manière, il est possible que des heuristiques rationnelles et comportementales puissent accéder en mode lecture/écriture à des objets-questions générés par d'autres heuristiques et puissent ainsi raisonner symboliquement à leur sujet, voire les altérer. Cette notion est au cœur de l'implémentation des biais cognitifs discutés à la section 5.

4.2. *Un modèle mental pour les agents*

L'implémentation de l'architecture *R&B* impose la définition préalable d'un modèle mental pour l'agent. L'architecture *R&B* est capable en théorie de supporter plusieurs types de modèles pourvu qu'ils soient exprimés dans le formalisme de $\mathscr{M}$. Ici, nous allons donc définir un modèle particulier qui, d'une part, est juste suffisamment simple pour supporter les exemples présentés à la section 5.2 et qui, d'autre part, couvre les principales notions que l'on peut trouver dans la littérature sur la modélisation des états mentaux (Sabater *et al.*, 2003). C'est pourquoi certaines simplifications ont été introduites, *e.g.* nous considérons les traits et les rôles comme étant statiques (alors que certains travaux dans le domaine des organisations considèrent que l'autorité peut évoluer – ceci est bien sûr une question d'échelle, et pour ce qui nous concerne, nous nous limitons à une session dialogique).

Nous distinguons quatre types d'états mentaux organisés selon deux axes : dynamicité et arité, comme illustré dans le tableau 2. Chaque état est défini par une valeur décimale dans $[-1.,1.]$, où $[0,1.]$ dénote l'intensité du concept, $[-1.,0]$ dénote l'intensité de son antonyme et 0 la position « neutre ».

Rôles (Ψ_R) ils représentent les relations statiques entre l'agent et les autres acteurs du monde, soit pour un AAC, l'usager. Nous définissons deux catégories principales :

– *Autorité* : il s'agit du droit que l'agent pense avoir de se comporter de manière directive vis-à-vis des autres et réciproquement de ne pas accepter les directives venant des autres. Ce rôle est souvent antisymétrique : `Authority(X,Y) = -Authority(Y,X)` où '-' dénote la relation antonyme.

– *Familiarité* : il s'agit du droit que l'agent pense avoir de pouvoir se comporter de manière informelle, voire familière, envers les autres. Ce rôle est souvent symétrique : `Familiarity(X,Y) = Familiarity(Y,X)`

Traits (Ψ_T) ils correspondent aux attributs de personnalité classiques qui sont considérés comme stables chez l'agent. Ils sont implémentés dans le modèle des cinq facteurs (appelé FFM ou encore OCEAN), utilisé couramment en psychologie (Goldberg, 1981) :

– **O**penness : attirance pour l'aventure, l'imagination, et la curiosité. Ce trait se divise en six facettes dans le standard NEO PI-R (discuté plus bas) : Fantasy, Aesthetics, Feelings, Actions, Ideas, Values ;

– **C**onscientiousness : tendance à l'autodiscipline et au désir d'atteindre et de réaliser les buts. Facettes : Competence, Orderliness, Dutifulness, Achievement-striving, Self-discipline, Deliberation ;

– **E**xtraversion : personnalité dominée par les émotions fortes et positives et une tendance à rechercher la compagnie des autres. Facettes : Warmth, Gregariousness, Assertiveness, Activity, Excitement-seeking, Positive-emotions ;

– **A**greeableness : tendance à se comporter de manière compassionnelle et coopérative. Facettes : Trust, Straightforwardness, Altruism, Compliance, Modesty, Tender-mindedness ;

– **N**euroticism : tendance à éprouver des sentiments et émotions négatifs. Facettes : Anxiety, Angry-Hostility, Depression, Self-consciousness, Impulsiveness, Vulnerability.

Dans des travaux récents, portant sur la classification des traits de personnalité (Bouchet *et al.*, 2010), nous avons été amenés à raffiner ce modèle de traits en utilisant les résultats issus de la littérature en psychologie, en particulier les *facettes* – ou *facets* du modèle NEO PI-R (Costa *et al.*, 1992) qui pour chaque trait proposent six sous-cas (chacun muni d'un pôle positif : le concept, et d'un pôle négatif : son antonyme), définissant ainsi un espace à 60 positions.

En fait, si on se place du point de vue purement computationnel, les définitions associées aux facettes de la taxonomie OCEAN/NEO PI-R manquent encore de précision. C'est pourquoi nous avons raffiné les facettes NEO PI-R en « schémas comportementaux », obtenus à partir de la classification de gloses WordNet, associées à des adjectifs de personnalité (Sansonnet *et al.*, 2010). Nous ne donnerons pas la liste de ces schémas (il y a en moyenne 2.3 schémas par facette) mais nous les utilisons implicitement dans les exemples 3 et 4 développés à la section 5.2.

Humeurs (Ψ_h) (en anglais *moods*) sont associées aux attributs mentaux de l'agent qui varient au cours du temps sous le contrôle des heuristiques et des biais, ceci en fonction des états mentaux précédents de l'agent ainsi que de l'état courant du monde. Nous définissons alors pour un agent :

– *énergie* : force physique ;

– *bonheur* (ou *happiness*) : bien-être physique ;

– *confiance en soi* : assurance cognitive ;

– *satisfaction* : satisfaction cognitive.

Les attributs liés à la corporalité de l'agent (Énergie, Bonheur) ne sont instanciés que dans les situations où l'agent est effectivement muni d'une corporalité (par exemple un personnage dans un jeu vidéo). Ils sont moins pertinents dans le cas des AAC et ne seront pas pris en compte ici.

Affects (Ψ_a) le terme affect est utilisé de manière spécifique ici pour dénoter les relations variables entre l'agent et l'usager. Nous distinguerons trois sortes d'affects :

– *dominance* : l'agent se sent puissant vis-à-vis d'un autre acteur. Cette relation est souvent antisymétrique : `Dominance(X,Y) = -Dominance(Y,X)` ;

– *coopération* : l'agent a tendance à se comporter de manière sympathique et coopérative avec les autres. Cette relation n'est pas nécessairement symétrique ;

– *confiance en l'autre – Trust* : l'agent a le sentiment qu'il peut faire confiance en l'autre (pas nécessairement symétrique).

Implémentation de l'état mental de l'agent dans le modèle $\mathscr{M}$: Le modèle de l'état mental de l'agent est dénoté de manière abrégée R_{af}-T_{ocean}-M_{hesc}-A_{dct} où nous représentons un attribut l'initiale du symbole de son nœud dans l'arbre décrit ci-dessous :

```
M.A.mind[
role[ // STATIQUE, RELATIONS SOCIALES A<->U
authority[-1..1],
familiarity[-1..1]
],
trait[ // STATIQUE, RELATIF AU MONDE
openness[-1..1],
conscientiousness[-1..1],
extraversion[-1..1],
agreeableness [-1..1],
neuroticism[-1..1],
],
mood[ // DYNAMIQUE, INTRINSEQUE A L'AGENT
happiness[-1..1]
energy[-1..1],
satisfaction[-1..1],
confidence[-1..1],
],
affect[ // DYNAMIQUE, RELATION AFFECTIVES A<->U
dominance[-1..1],
cooperation[-1..1],
trust[-1..1]
]]
```

Un attribut peut être accédé par son chemin complet comme par exemple « $\mathscr{M}.\mathscr{A}$.mind.mood.happiness.value » ou dans la notation abrégée M_h pour R_{af}-T_{ocean}-M_{hesc}-A_{dct} si on reprend l'exemple de l'attribut bonheur (*happiness*).

Intervalles mentaux : dans le modèle formel décrit ci-dessus, les valeurs des attributs v sont des nombres décimaux $\in [-1., 1.]$, mais il peut être aussi utile de considérer une échelle à cinq positions symboliques bâtie sur une partition du domaine $[-1., 1.]$ en cinq intervalles contigus :

$v \in [-1., -0.8[$	<	fortement antonymique
$v \in [-0.8, -0.2[$	-	antonymique
$v \in [-0.2, 0.2[$	=	neutre
$v \in [0.2, 0.8[$	+	positif
$v \in [0.8, 1.]$	>	fortement positif

Notons qu'afin de conserver une certaine simplicité dans le modèle, nous avons retenu des intervalles discrets mais la logique floue pourrait apporter une segmentation plus appropriée. Les valeurs sont notées par exemple, $M_h^+ \wedge A_c^<$ pour dire que l'agent est heureux et complètement antagoniste ; plusieurs intervalles peuvent être unis en juxtaposant les signes derrière l'attribut : *e.g.* $M_h^{+>}$ veut dire « heureux ou très heureux ».

Évènements mentaux : un phénomène significatif concerne la dynamique des valeurs des attributs mentaux (humeurs et affects). Lorsque la valeur d'un attribut est modifiée par une question (*e.g.* elle est incrémentée ou décrémentée) elle peut passer d'un intervalle à son voisin : l'intervalle précédent (nous dirons vers le bas) ou l'intervalle suivant (*resp.* vers le haut). Chaque fois qu'une transition d'état mental est ainsi franchie, cela produit un évènement noté $\backslash$ (*resp.* $/$) pour une transition vers le bas (*resp.* vers le haut), qui est attaché à la question ayant provoqué cette transition.

Par exemple, $Q_i^+ \wedge \backslash M_h^-$ veut dire que la question Q_i a été exécutée avec succès (pour +/- *cf.* encadré **Exemple 1** section 4.3) et que la valeur de l'attribut happiness de l'agent a effectué une transition de $M_h^{=+>}$ vers M_h^-. Du fait que les évènements sont attachés à la question qui les a déclenchés, ils sont effacés quand la question est achevée. Cela empêche l'agent d'entrer dans une boucle infinie ; par exemple au lieu de répéter M_h^- (« Je suis triste ») il dit une seule fois $\backslash M_h^-$ (« Je me sens triste (tout à coup) »).

4.3. *Les heuristiques*

Une heuristique définit une réaction rationnelle ou comportementale pour une classe de requêtes formelles, exprimées par des expressions en FRL. D'une certaine manière, on peut dire qu'une heuristique « exprime en dur » la réaction de l'agent. La classe des expressions FRL prises en compte par une heuristique est définie par une expression de *pattern matching* (FRL-pattern) portant sur le langage FRL. La forme générale d'une heuristique est alors donnée par une expression de la forme :

$\mathsf{H} : \mathsf{id}[\mathsf{FRL\text{-}pattern}] :\rightarrow \{\mathsf{GScript}_1, \ldots, \mathsf{GScript}_n\}$

Avec les définitions :

$$
\begin{array}{lll}
\mathsf{GScript} & \equiv & \{\mathsf{Guard}_1 \rightarrow \mathsf{Script}_1, \ldots, \\
 & & \mathsf{Guard}_n \rightarrow \mathsf{Script}_n\} \\
\mathsf{Guard}_i & \equiv & \text{logical expr.} \mid \emptyset \qquad (\emptyset = \text{True}) \\
\mathsf{Script}_i & \equiv & \mathsf{Instr} \mid \{\mathsf{Instr}_1, \ldots, \mathsf{Instr}_n\} \\
\mathsf{Instr}_i & \equiv & \text{basic oper.} \mid \text{question MQL} \mid \mathsf{GScript} \\
\text{question MQL} & \equiv & \mathsf{Q}[\text{Question-id, MQL expr.}]
\end{array}
$$

Dans la mesure où les instructions peuvent comporter des imbrications récursives de scripts gardés, il est possible de définir des structures conditionnelles permettant de programmer des arbres de décision, telles que celles qui apparaissent souvent dans l'implémentation des heuristiques.

Algorithme 1. SH : procédure générique de scheduling des heuristiques

$req \leftarrow$ la requête FRL issue de la file d'attente de la NLP-chaîne de $\mathscr{I}$
$H_r \leftarrow$ la liste ordonnée des heuristiques du moteur rationnel $\mathscr{E}_r$
$H_b \leftarrow$ la liste ordonnée des heuristiques du moteur comportemental $\mathscr{E}_b$
$H_{rb} \leftarrow H_r || H_b$
$H^*_{rb} \leftarrow match(H_{rb}, req)$ — liste ordonnée des heuristiques qui matchent req
$active \leftarrow True$
tant que $active$ **faire**
  $active \leftarrow False$
  si $mode = deterministe$ **alors**
    pour tout $h_i \in H^*_{rb}$ **faire**
      sachant que h_i est une liste ordonnée de règles $r_j = \langle g_j \rightarrow s_j \rangle$
      pour tout :2 $r_j \in h_i$ **faire**
        si la garde g_j est vide ou s'évalue à $True$ **alors**
          Exécuter le script associé s_j ; Enlever r_j de h_i ; $active \leftarrow True$; **exit for all :2**
        finsi
      fin pour
    fin pour
  sinon si $mode = stochastique$ **alors**
    pour tout $h_i \in H^*_{rb}$ **faire**
      sachant que h_i est un ensemble de règles $r_j = \langle g_j \rightarrow s_j \rangle$
      $L_{hit} \leftarrow \emptyset$ — récolte la liste de toutes les règles couramment actives
      pour tout $r_j \in h_i$ **faire**
        si la garde g_j est vide ou s'évalue à $True$ **alors**
          Ajouter r_j à L_{hit}
        finsi
      fin pour
      si $L_{hit} \neq \emptyset$ **alors**
        une règle r^* est tirée au sort dans L_{hit} ; $active \leftarrow True$;
        Exécuter le script associé s^* — r^* n'est pas enlevé de h_i
      finsi
    fin pour
  sinon si $mode =$ etc. **alors**
    Autres politiques de scheduling.
  finsi
fin tant que

Scheduling des heuristiques pour une requête : quand une requête FRL est envoyée à l'agent, elle est placée dans la file d'attente pour être traitée, à son tour, par la procédure générique de scheduling des heuristiques rationnelles et comportementales (notée SH) dont le principe est donné dans l'algorithme 1. Le scheduler d'heuristiques prend une requête *req* dans la file et déclenche les heuristiques rationnelles et comportementales qui matchent *req* et il assure le coroutinage de leur exécution. Le grain d'exécution du coroutineur est déterminé par les scripts qui s'exécutent de manière insécable. Plusieurs modes/stratégies de scheduling sont alors possibles : les deux modes de base sont le mode déterministe et le mode stochastique, décrits dans l'algorithme 1 ; d'autres modes sont programmables, comme par exemple des variantes des modes proposés par défaut. La terminaison de l'algorithme dépend du mode sélectionné : inactivabilité ou exhaustion des règles dans le cas déterministe ; *resp.* inactivabilité dans le cas stochastique. En cas de mauvaise programmation des heuristiques, SH peut boucler si un script boucle, ou encore si une garde demeure infiniment active (cas stochastique seulement, où il ne faut pas utiliser de gardes vides).

Exemple 1 : une réaction rationnelle simple Supposons que l'usager pose la demande suivante : « Quel est ton âge ? » qui se traduit en FRL par ASK_u[agent.age]. Celle-ci peut être traitée par une heuristique telle que celle donnée et expliquée dans l'encadré **Exemple 1** (mode sélectionné : déterministe).

Exemple 1 : Heuristique rationnelle pour les demandes sur la valeur d'un attribut de l'agent.

1: H_r : ask-agent-attribute[ASK_u[agent.x_]]:$\rightarrow$ {
2: $\rightarrow Q$[i, GET[x_]],
3: $Q_i^+ \rightarrow Q$[j, SET[$\mathscr{R}$.reply, TELL_a[agent.x_, Q_i.value]]],
4: $Q_i^- \rightarrow Q$[j, SET[$\mathscr{R}$.reply, TELL_a[Q_i.failure]]]
5: }

Explications :

1: x_ est une variable de pattern matching qui capture tout symbole d'attribut comme âge, sexe, nom...
2: une garde vide oblige le script à être exécuté immédiatement avec x_ valant « age » alors age sera un raccourci pour le chemin complet M.A.age.value dans $\mathscr{M}$ (sous-arbre value[v] $\Rightarrow$ v)
i est un identifiant unique utilisé ensuite pour référer l'objet-question par Q_i
3: Q_i^+ est un raccourci pour Q_i.status == done $\wedge$ Q_i.head == OK
TELL_a[agent.x_, Q_i.value] est une réponse en FRL pour l'usager
4: Q_i^- est un raccourci pour Q_i.status == done $\wedge$ Q_i.head == FAIL
Q_i.failure renvoie la cause de l'échec (*e.g.* NOTFOUND[path, concept])

Exemple 2 : une réaction comportementale simple Supposons que l'usager tape : « Je te déteste ! ». Cela va produire la requête FRL DISLIKE_u[agent]. Une heuristique comportementale possible est alors celle donnée dans l'encadré **Exemple 2** (mode sélectionné : déterministe).

Exemple 2 : Heuristique comportementale pour les remarques négatives sur l'agent.

```
1:  H_b : dislike-agent[DISLIKE_u[agent]]:→ {
2:                                  →  {   Q[i, MAP[energy, λ x.x∗0.9]],
3:                                         Q[j, MAP[confidence, λ x.x∗0.9]],
4:                                         Q[k, MAP[cooperation, λ x.x∗0.9]],
5:                                     }
6:    Q_i^+ ∧ Q_i.value < −0.5     →       ADD[R.reply, TELL_a[energy, 'tired']]
7:    Q_j^+ ∧ Q_j.value < −0.5     →       ADD[R.reply, TELL_a[confidence, 'depressed']]
8:  }
```

Explications :

- `2:` l'agent exécute une séquence de trois questions qui modifient son état mental. energy abrège le chemin M.A.mind.mood.energy.value dans le modèle de l'agent. $\lambda x.x*0.9$ est une λ-expression décrémentant son argument x de 10%.
- `6:` si l'énergie est très basse, on ajoute « Je me sens fatigué » à la liste des choses à dire.
- `7:` idem pour la Confidence ; le coroutinage assure qu'on puisse avoir les deux en même temps « Je me sens fatigué *et* déprimé ».

5. Étude de cas : les biais cognitifs

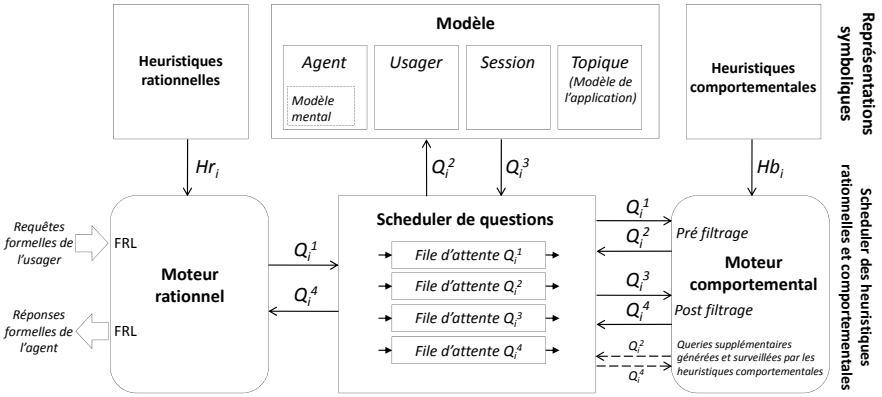
5.1. *Principe des biais cognitifs*

Notre premier objectif est de développer un outil de modélisation capable de supporter et d'expérimenter plusieurs hypothèses différentes concernant la nature des relations entre les comportements rationnels et les comportements psychologiques qui peuvent impliquer des stratégies de scheduling distinctes. Pour cela, il faut que l'algorithme de scheduling soit capable d'intégrer de nouveaux modes (*cf.* algorithme 1) et que sa structure de gestion des questions soit paramétrable. C'est pourquoi nous avons proposé, à titre d'exemple d'hypothèse à étudier, le scénario des biais cognitifs (BC) qui postule que le moteur rationnel et le moteur comportemental 1) sont construits de manière indépendante par des concepteurs indépendants et 2) que ces moteurs fonctionnent de manière indépendante.

– Le moteur rationnel $\mathscr{E}_r$ est un ensemble d'heuristiques rationnelles H_{r_i} qui s'appliquent aux requêtes de l'utilisateur, afin de les résoudre. $\mathscr{E}_r$ produit des questions MQL Q_i sur le modèle $\mathscr{M}$ et réagit aux rapports rendus par ces questions.

– Le moteur comportemental $\mathscr{E}_b$ est un ensemble d'heuristiques comportementales H_{b_i} (appelées dans ce cas, Biais Cognitifs ou simplement Biais) qui s'appliquent aux questions Q_i produites par $\mathscr{E}_r$. Étant donnée une question Q_i, $\mathscr{E}_b$ peut à différents moments de la résolution de la question : 1) l'altérer et/ou 2) altérer son rapport et/ou réagir à la question Q_i en modifiant $\mathscr{MA}$.A.mind.(mood/affect). Ce processus est appelé le *filtrage de questions*.

Dans le scénario BC, $\mathscr{E}_r$ n'a pas connaissance de l'existence et des effets de $\mathscr{E}_b$ sur les questions. De plus, $\mathscr{E}_r$ possède un accès en mode lecture/écriture à l'ensemble des éléments de $\mathscr{M}$ *sauf* au sous-arbre $\mathscr{MA}$.mind (par exemple, il n'est pas possible d'utiliser l'expression M_{ht}^{+} dans une garde d'une heuristique de $\mathscr{E}_r$). De même, $\mathscr{E}_b$



Q_i^1 La question Q_i^1 a été produite par le moteur rationnel $\mathscr{E}_r$. Elle entre dans la file d'attente notée Q_i^1 du scheduler pour être traitée par le moteur comportemental $\mathscr{E}_b$.

Q_i^2 Lorsqu'elle a subit un pré-filtrage par $\mathscr{E}_b$, elle passe dans l'état 2 et entre dans la file d'attente Q_i^2 du scheduler en attente d'être traitée par le moteur rationnel qui va la résoudre dans le modèle $\mathscr{M}$ (impliquant des lectures et/ou écritures sur $\mathscr{M}$) .

Q_i^3 La question pré-filtrée a été résolue sur $\mathscr{M}$ et ses attributs-résultat sont remplis : elle passe dans l'état 3 et entre dans la file d'attente Q_i^3 du scheduler en attente d'être traitée à nouveau par $\mathscr{E}_b$.

Q_i^4 La question a été post-filtrée par $\mathscr{E}_b$. Elle passe dans l'état 4 et entre dans la file d'attente Q_i^4 du scheduler où elle est à nouveau disponible pour le traitement par des expressions de garde et de script de $\mathscr{E}_r$.

Note : Les scripts exécutés dans $\mathscr{E}_b$ ou $\mathscr{E}_r$ peuvent à leur tour engendrer des questions additionnelles (états 2 ou 4 en pointillés), soumises au même processus de filtrage.

Figure 3. *Architecture R&B avec son scheduler dédié au scénario des biais cognitifs*

possède un accès en mode lecture/écriture aux éléments de $\mathscr{MA}$.mind.(mood/affect) et n'a qu'un mode lecture sur les autres éléments de $\mathscr{M}$. En conséquence, $\mathscr{E}_b$ n'a pas connaissance du processus de résolution rationnelle en cours (la tâche globale), qui est du ressort de $\mathscr{E}_r$. De plus, $\mathscr{E}_b$ n'est activé que lorsque le scheduler lui demande de filtrer des questions produites par $\mathscr{E}_r$.

L'implémentation d'un tel fonctionnement s'appuie sur la procédure générique de scheduling SH (*cf.* algorithme 1), à laquelle nous apportons trois spécifications :

1) le mode déterministe est sélectionné ;

2) un attribut supplémentaire, « .status », est introduit dans la structure de données des questions, dénoté ensuite Q_i^{status}, afin de représenter quatre états successifs de son filtrage (status $\in [1,4]$) ;

3) la file d'attente de SH est décomposée en quatre files, associées aux quatre états. On peut suivre ces quatre états dans la figure 3 qui illustre l'architecture générale *R&B* et son instance spécifique pour l'étude des biais cognitifs.

5.2. *Implémentation de comportements au moyen des biais cognitifs*

Un comportement H_b décrit une réaction de l'agent en fonction de son état mental courant (*i.e.* des valeurs des feuilles de $\mathscr{M}\mathscr{A}$.mind).

Exemple 3 : supposons l'existence d'un agent qui, à un moment donné, possède un haut niveau de satisfaction et n'est pas neurotique $M_s^{>} \wedge T_n^{<-=}$ (*i.e.* qui peut être situé par exemple aux positions Conscientiousness/Self-confidence+ et Neuroticism- dans la taxonomie des traits OCEAN/NEO-PI R définie à la section 4.2). Il aura sans doute tendance à sous-estimer certaines phrases négatives provenant de l'usager. Supposons de plus que l'usager dise qu'il/elle déteste l'agent :

« Je te déteste » $\Rightarrow$ DISLIKE$_u$[agent].

Une certaine heuristique rationnelle enregistrera alors « froidement » cette information en ajoutant l'agent à la liste des non-préférences (*dislikes*) de l'usager :

H_r : dislike-agent[DISLIKE$_u$[agent]] :$\rightarrow$ {$\ldots \rightarrow Q^1$[i, `ADD[`$\mathscr{U}$.dislikes, agent`]`], $\ldots$}

Dès lors, un biais possible sur cette question produite par $\mathscr{E}_r$ dans l'état 1 est donné et expliqué dans l'encadré **Exemple 3**.

Exemple 3 : Biais de perception positif sur une remarque négative.

```
1:    H_b : see-life-in-pink[Q^1[i_, ADD[(A |U).dislikes, y_]]:→ {
2:      M_s^> ∧ T_n^<-=    →    Q^2[i, VOID[]]
3:    ...}
```

Explications :

1: L'agent refuse de considérer une question ajoutant une non-préférence dans $\mathscr{M}$.

2: Donc il la remplace par une question vide `VOID[]` qui est produite avec l'état = 2. Remarquons qu'un rapport de succès `OK[]` sera retourné à H_r qui est en attente d'une condition de la forme Q_i^{+4} (+ est un succès et état = 4) ce qui fait que le processus rationnel continuera sans que H_r sache qu'un biais a été appliqué

Exemple 4 : un agent systématiquement malheureux et neurotique $T_n^{+} \wedge M_h^{-}$ (*i.e.* qui peut être situé par exemple à la position Neuroticism/Depression+ dans la taxonomie des traits OCEAN/NEO-PI R, section 4.2) a probablement tendance à percevoir une « négativité » excessive dans tout ce que lui dit l'usager. Supposons que l'usager dise :

« J'aime la couleur du titre » $\Rightarrow$ LIKE$_u$[gui.title.color].

Alors il y aura une certaine heuristique rationnelle qui enregistrera « froidement » cette entité à la liste des préférences de l'usager (de manière similaire à l'exemple 3) et qui de plus positionnera un certain attribut de cette entité à *high*. Cet attribut, dénoté *wfu* (worth-for-user), représente le fait que l'usager a indiqué que cette entité a de la valeur, esthétique ou autre, pour lui.

H_r : like-entity[LIKE$_u$[x_]] :$\rightarrow$ {$\ldots \rightarrow Q^1$[i, `SET[`x_.wfu, 'high'`]`], $\ldots$}

Cependant l'agent peut réagir par un comportement négatif à cette sorte de question et un biais implémentant cela est donné et expliqué dans l'encadré **Exemple 4**.

Exemple 4 : Biais de perception négatif sur une remarque positive.

```
1:    H_b : see-life-in-black[Q^1[i_, SET[x_.wfu, 'high']]]:→ {
2:      M_s^> ∧ T_n^<−=    →    {
3:                              Q^2[i, SET[x_.wfu, 'high'],
4:                              ADD[ℛ.reply, REQUEST_a[JUSTIFY_u[ℳ.ℛ ]]]
5:                              Q^2[j, MAP[confidence, λ x.x*0.8]
6:                              ...},
7:    ...}
```

Explications :

3 : La question elle-même n'est pas altérée. Elle progresse juste de l'état 1 à 2.

4 : Une réplique fielleuse est ajoutée à $\mathscr{R}$.reply pour demander à l'usager de se justifier.

5 : De plus, parce qu'il est déprimant de percevoir la bonne humeur des autres quand soi-même on est de mauvaise humeur, il décrémente son attribut confidence de 20%.

Maintenant on peut aussi supposer qu'on a défini un autre biais qui filtre les questions dans l'état 3 et que ce biais est associé à un évènement mental qui aurait pu être déclenché par une mise à jour de l'état mental, comme la mise à jour définie à la ligne 5 de l'exemple 4, par exemple $\backslash M_c^-$:

H_b : on-entering-unconfidence[Q^3[i_, MAP[confidence, f_]]]:→ {$Q_i^{+4} \wedge \backslash M_c^- \rightarrow$ ADD[$\mathscr{R}$.reply, TELL$_a$[$\backslash M_c^-$]] }

Ceci peut alors avoir comme effet d'ajouter l'expression « Je me sens déprimé d'un seul coup » à la réponse globale de l'agent, dans le cas où l'agent entre vers le bas dans l'intervalle M_c^- et ainsi de suite. Remarquons que les questions supplémentaires produites par les biais sont dans l'état 2, ce qui fait qu'elles peuvent être à leur tour biaisées (lorsqu'elles sont dans l'état 3 ou 4) par d'autres biais, similaires à celui décrit ci-dessus, au moment où elles sont renvoyées à l'usager. Encore une fois, H_r n'a pas connaissance de ces biais et il ajoutera ensuite un rapport OK[] à $\mathscr{R}$.reply. Finalement le module d'expression en langue naturelle produit une réplique comme « D'accord mais pouvez vous justifier cela ! ? » voire même « J'en prends note, mais expliquez moi pourquoi ? ... D'un seul coup, tout ça me déprime ».

6. Discussion

6.1. *Implémentation*

Les propositions présentées dans la section 3, s'appuient sur des outils logiciels développés dans le cadre du projet d'agent conversationnel assistant DAFT [6] ; en particulier pour la chaîne de traitement des requêtes en langue naturelle ainsi que la définition du langage FRL décrit à la section 3.3. Toutefois le modèle d'agent proposé

6. Documentation listée à : `http://www.limsi.fr/~jps/research/daft/index.html`

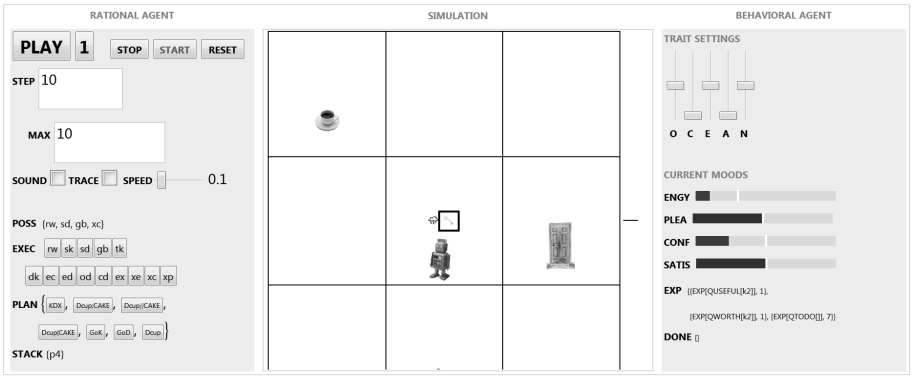


Figure 4. *Tableau de bord du toolkit R&B présentant une étude de cas. À gauche, les contrôles permettent d'exécuter des plans ou de simples actions en paramétrant différents éléments comme la vitesse, le nombre de pas de la simulation etc. Le panneau indique : la liste des actions possibles à chaque instant, l'action courante, le plan courant ainsi que la pile des plans (en bas). À droite, on peut positionner une personnalité pour l'agent, dans le modèle psychologique OCEAN, au moyen de curseurs et les barres indiquent alors les valeurs (entre -1 et 1) des états physiques et mentaux de l'agent, respectivement l'énergie physique le plaisir physique, la confiance en soi, la satisfaction intellectuelle. La liste des attentes de l'agent (états désirés) est aussi donnée (notée EXP). Au centre, la simulation se déroule en pas à pas ou en continu*

à la section 4 dépasse largement le cadre de DAFT et fait l'objet d'une proposition nouvelle, l'architecture *R&B*. Bien que s'appuyant sur les modules de DAFT cités ci-dessus, celle-ci nécessite des développements logiciels spécifiques. Une première version de l'architecture *R&B* a été implémentée en Mathematica. Le toolkit d'essai est accessible et téléchargeable sur la page du projet *R&B* [7].

La figure 4 illustre le tableau de bord de ce toolkit avec une expérience où un robot doit exécuter des plans rationnels (contrôlés dans la partie gauche) sous l'influence de traits de personnalité statiques (ici FFM dénotée OCEAN — contrôlés dans la partie droite-haut) et d'états mentaux dynamiques dont l'évolution au cours de l'exécution du plan (choisi dans le panneau de gauche) sont affichés dans la partie droite-bas.

A terme, cet outil doit permettre de définir un certain nombre de scénarios expérimentaux ayant pour but d'être présentés à des sujets humains. Ceux-ci devraient alors noter la perception effective qu'ils ont des comportements de l'agent au niveau de sa compétence (rationalité) puis la mettre en contraste avec la perception qu'ils ont des comportements trahissant certaines caractéristiques psychologiques exhibées par l'agent. La comparaison entre les traits perçus et les traits implémentés devrait per-

7. `http://www.limsi.fr/~jps/research/rnb/rnb.htm`

mettre d'effectuer une première évaluation du modèle au niveau de la modélisation cognitive.

Les architectures servant de support à *R&B* (DAFT, DIVA) ont déjà été utilisées pour effectuer des validations expérimentales. Par exemple, à la section 2.3, nous avons donné un exemple reposant sur DIVA. L'architecture *R&B*, descendant de DAFT et DIVA, est donc capable d'implémenter, au moyen des heuristiques, des comportements tels que les trois types de répétitions mentionnées. Les résultats déjà obtenus à partir des expérimentations avec DIVA (Xuetao *et al.*, 2009) donnent une première indication sur le fait que les sujets placés face à des agents *R&B* a) percevront les comportements b) les percevront comme ayant une source psychologique et c) les percevront comme « de type humain » (*human-likeness*).

L'architecture *R&B* repose sur l'utilisation de règles explicites, comme pour SOAR ou ACT-R. De tels systèmes ont été classiquement critiqués pour leur difficulté à prendre en compte et à gérer de grandes quantités de règles. Notre étude porte sur des études de cas qui ne souffrent pas de ce problème et ont même l'avantage d'expliciter l'intrication des relations entre rationalité et psychologie. Toutefois cela ne peut être qu'une étape vers une démarche plus exhaustive où les règles seraient exhibées de manière systématique 1) par des mécanismes d'apprentissage ou bien 2) par des mécanismes génériques. Nous nous orientions vers la seconde solution dans nos derniers travaux (Bouchet *et al.*, 2011).

6.2. *Travaux connexes*

Le positionnement de nos travaux par rapport aux études récentes qui ont été présentées à la section 2 se situe essentiellement à trois niveaux :

1) Direction des influences : dans leurs travaux, Gratch et Marsella ont étudié surtout les phénomènes d'*appraisal* (Scherer *et al.*, 2001) et de *coping* (Lazarus, 1966) qui portent sur les influences des événements, issus des actions dans le monde, sur la psychologie des agents. Ainsi, le modèle que nous proposons à la section 3 étend ceci à l'influence de la psychologie sur le comportement rationnel. D'autres travaux s'appuient sur l'influence des traits de personnalité sur diverses parties des architectures d'agents conversationnels. Par exemple, Malatesta *et al.* (Malatesta *et al.*, 2007) utilisent les traits de personnalité pour influencer l'expression des émotions, en particulier par leur impact sur les processus d'*appraisal* et de *coping* à l'intérieur d'une architecture de type OCC (Ortony *et al.*, 1988). Cependant, dans les modèles fondés sur OCC, les influences vont du monde physique vers le mental de l'agent (réaction à des événement externes) et non du mental de l'agent vers ses actions sur le monde physique. La même remarque s'applique à l'architecture FATIMA de Paiva *et al.* (Doce *et al.*, 2010).

2) Prise en compte de notions fondées psychologiquement : dans l'architecture de référence que représente CoJACK (*cf.* section 2.2) pour le domaine BDI, nous avons vu qu'il s'agit d'émuler certaines caractéristiques de la physiologie ainsi que des capa-

cités cognitives humaines (*e.g.* la mémoire limitée) et non de la psychologie. D'autres travaux comme l'architecture PMFserv (Silverman *et al.*, 2006) ont une approche similaire : les auteurs mettent aussi l'accent sur l'influence de la physiologie et des capacités cognitives sur le cycle de délibération de l'agent, dans un cadre d'applications de simulations militaires, fortement liées à l'action, mais pas sur les aspects proprement psychologiques.

3) Prise en compte de la dynamique du cycle de délibération : Rizzo *et al.* font le pont entre architecture BDI et agents conversationnels (Rizzo *et al.*, 1999) et prouvent que les buts et les plans peuvent être utilisés pour représenter la personnalité, en attribuant des comportements spécifiques (c'est-à-dire directement liés aux traits de personnalité) dans le processus de délibération BDI. Par exemple, les traits de personnalité (*cf.* section 4.2) sont utilisés pour choisir entre des buts alternatifs (*i.e.* les traits influencent les désirs). Cependant on notera qu'une fois choisis, les buts sont planifiés et exécutés directement, sans plus d'influence de la psychologie, alors que dans le modèle que nous proposons, les traits de personnalité influencent des buts déjà engagés (*i.e.* chez nous, les traits influencent les intentions).

7. Conclusion

Dans le contexte des nouveaux outils d'assistance pour le grand public nous avons vu que la fonction d'assistance peut être supportée aussi bien par le processus de raisonnement rationnel sur la tâche en cours que par le processus comportemental qui prend en compte les aspects conversationnels entre l'agent et l'usager. Cependant, la nature des relations, parfois imbriquées, entre le raisonnement rationnel et le raisonnement comportemental demeure une question ouverte. Ceci nous a amenés à considérer l'étude d'un cadre de modélisation dédié pour mener des formalisations ainsi que des expérimentations sur diverses hypothèses : les agents *R&B*. L'architecture *R&B* permet une implémentation flexible des scénarios qu'on veut tester à propos des relations entre les heuristiques rationnelles et comportementales. Comme première étape de validation de cette architecture, nous avons implémenté une étude de cas simple mais qui met bien à l'épreuve notre cadre, les biais cognitifs, où les deux moteurs ($\mathscr{E}_r$ et $\mathscr{E}_b$) sont conçus de manière indépendante et fonctionnent de manière indépendante, mais cependant interagissent grâce au partage d'objets internes au modèle (les questions).

Nos prochains travaux, outre le développement de la programmation du toolkit *R&B* lui-même, vont porter sur l'utilisation du toolkit d'agents assistants conversationnels pour le web (DIVA) que nous avons aussi développé, en conjonction avec le toolkit *R&B*, afin de mener des expérimentations avec des sujets humains sur divers scénarios comportementaux pour évaluer la perception effective que ces sujets ont des agents *R&B*.

8. Bibliographie

Allen J. F., Schubert L. K., Ferguson G., Heeman P., Hwang C. H., Kato T., Light M., Martin N., Miller B., Poesio M., Traum D. R., « The TRAINS project : a case study in building a conversational planning agent », *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 7, n° 1, p. 7-48, 1995.

Anderson J. R., « ACT : A simple theory of complex cognition », *American Psychologist*, vol. 51, n° 4, p. 355-365, 1996.

Bates J., « The role of emotion in believable agents », *Commun. ACM*, vol. 37, n° 7, p. 122-125, July, 1994.

Bevacqua E., de Sevin E., Pelachaud C., McRorie M., Sneddon I., « Building credible agents : Behaviour influenced by personality and emotional traits », *Proc. of the Int. Conf. on Kansei Engineering and Emotion Research (KEER'10)*, Paris, France, 2010.

Bouchet F., « Characterization of Conversational Activities in a Corpus of Assistance Requests », *in* T. Icard (ed.), *Proc. of the 14th Student Session of the European Summer School for Logic, Language, and Information (ESSLLI)*, Bordeaux, France, p. 40-50, July, 2009.

Bouchet F., Sansonnet J. P., « Tree-kernel and Feature Vector Methods for Formal Semantic Requests Classification », *in* P. Perner (ed.), *Machine Learning and Data Mining in Pattern Recognition*, IBaI Publishing, Leipzig, Germany, p. 126-140, July, 2009.

Bouchet F., Sansonnet J. P., « Classification of Wordnet Personality Adjectives in the NEO PI-R Taxonomy », *Fourth Workshop on Animated Conversational Agents, WACA 2010*, Lille, France, p. 83-90, 2010.

Bouchet F., Sansonnet J. P., « Influence of personality traits on the rational process of cognitive agents », *The 2011 IEEE/WIC /ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Lyon, France, 2011.

Bratman M. E., *Intentions, Plans, and Practical Reason*, Harvard University Press, Cambridge, MA, 1987.

Bratman M. E., Israel D. J., Pollack M. E., « Plans and Resource-bounded Practical Reasoning », *Computational Intelligence*, vol. 4, n° 4, p. 349-355, 1988.

Buisine S., Martin J., « Children's and Adults' Multimodal Interaction with 2D Conversational Agents », *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, Portland, Oregon, USA, p. 1240-1243, April, 2005.

Caelen J., Xuereb A., *Interaction et pragmatique : Jeux de dialogue et de langage*, Lavoisier - Hermes, 2007.

Capobianco A., Carbonell N., « Contextual online help : elicitation of human experts' strategies », *Proceedings of HCI'01*, New Orleans, p. 824-828, August, 2001.

Carbonell N., « Towards the design of usable multimodal interaction languages », *Universal Access in the Information Society*, vol. 2, n° 2, p. 143-159, June, 2003.

Casali A., Godo L., Sierra C., « Graded BDI models for agent architectures », *Proceedings of CLIMA-V*, vol. 3487 of *Lecture Notes in Computer Science*, Lisbon, Portugal, p. 126-143, 2004.

Cassell J., Bickmore T., Billinghurst M., Campbell L., Chang K., Vilhjálmsson H., Yan H., « Embodiment in conversational interfaces : Rea », *CHI'99 : Proceedings of the SIGCHI conf. on Human factors in comp. syst.*, ACM Press, New York, NY, USA, p. 520-527, 1999.

Cassell J., Sullivan J., Prevost S., Churchill E. (eds), *Embodied Conversational Agents*, MIT Press, April, 2000.

Cohill A. M., Williges R. C., « Retrieval of Help Information for Novice Users of Interactive Computer Systems », *Human Factors*, vol. 27, n° 3, p. 335-343, 1985.

Coiro J., « Reading Comprehension on the Internet : Expanding Our Understanding of Reading Comprehension to Encompass New Literacies. », *The Reading Teacher*, vol. 56, n° 5, p. 458-465, 2003.

Cooper R., Engdahl E., Larsson S., Ericsson S., « Accommodating questions and the nature of QUD », *Proceedings of Gotalog 2000*, Gotheburg, p. 57-62, 2000.

Costa P. T., McCrae R. R., *The NEO PI-R professional manual*, Odessa, FL : Psychological Assessment Resources, 1992.

Damasio A. R., « Descartes Error : Emotion, Reason and the Human Brain », Putnam's Sons, New York : G.P., 1994.

Davis F. D., « Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology », *MIS Quarterly*, vol. 13, n° 3, p. 319-340, September, 1989.

Doce T., Dias J., Prada R., Paiva A., « Creating individual agents through personality traits », *Intelligent Virtual Agents (IVA 2010)*, vol. 6356 of *LNAI*, Springer-Verlag, Philadelphia, PA, p. 257-264, 2010.

Ekman P., Friessen W. V., Ellsworth P. C., *Emotion in the human face*, Pergamon Press, May, 1972.

Evertsz R., Ritter F. E., Busetta P., Pedrotti M., « Realistic Behaviour Variation in a BDI-based Cognitive Architecture », *Proc. of SimTecT'08*, Melbourne, Australia, 2008.

Ferguson G., Allen J. F., « TRIPS : an integrated intelligent problem-solving assistant », *AAAI'98/IAAI'98 : Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, American Association for Artificial Intelligence, Menlo Park, CA, USA, p. 567-572, 1998.

Fogg B. J., Soohoo C., Danielson D. R., Marable L., Stanford J., Tauber E. R., « How do users evaluate the credibility of Web sites ? : a study with over 2,500 participants », *Proc. of the 2003 conference on Designing for user experiences*, DUX '03, ACM, New York, NY, USA, p. 1-15, 2003.

Frijda N. H., *The Laws of Emotion*, Psychology Press, 2006.

Frijda N. H., Ekman P., Scherer K. R., *The Emotions*, Studies in Emotion & Social Interaciton, Cambridge University Press, 1986.

Galluccio R. G. P., « Humanizing CALL : The use of pedagogical agents as language tutors », *NERALLT 2006*, Cambridge, MA, USA, October, 2006.

Galon D., *The Savvy Way to Successful Website Promotion : Secrets of Successful Websites, Attracting On-Line Traffic, the Most Up to Date Guide to Top Positioning on Search Engines*, Trafford on Demand Pub, 1999.

Goldberg L. R., « Language and individual differences : The search for universal in personality lexicons », *Review of personality and social psychology*, vol. 2, p. 141-165, 1981.

Goodwin C., Heritage J., « Conversation Analysis », *Annual Review of Anthropology*, vol. 19, p. 283-307, 1990.

Gratch J., Marsella S., « A Domain-independent Framework for Modeling Emotion », *Journal of Cognitive Systems Research*, vol. 5, n° 4, p. 269-306, 2004.

Gratch J., Marsella S., « Evaluating a Computational Model of Emotion », *Autonomous Agents and Multi-Agent Systems*, vol. 11, n° 1, p. 23-43, 2005.

Grau B., Sabah G., Vilnat A., « Pragmatique et dialogue homme-machine », *Techniques et Sciences Informatiques*, vol. 13, n° 1, p. 9-30, 1994.

Grosz B., Sidner C. L., « Attention, intentions, and the structure of discours », *Computational Linguistics*, vol. 12, n° 3, p. 175-204, 1986.

Gurevich Y., « Evolving algebras 1993 : Lipari guide », *Specification and validation methods*, Oxford University Press, Inc., p. 9-36, 1995.

Hayes-Roth B., « What makes characters seem life-like ? », *In : Life-like Characters. Tools, Affective Functions and Applications*, Springer, Heidelberg, 2004.

Hayes-Roth B., Brownston L., van Gent R., « Multiagent collaboration in directed improvisation », *Proc. of the First Int. Conf. on Multi-Agent Systems ICMAS 95*, San Francisco, CA, p. 148-154, 1995.

Hobbs J., « Resolving pronoun references », *Lingua*, vol. 44, p. 311-338, 1978.

John O. P., Robins R. W., Pervin L. A. (eds), *Handbook of Personality : Theory and Research*, 3rd edn, The Guilford Press, August, 2008.

Kopp S., Wachsmuth I., « Synthesizing multimodal utterances for conversational agents », *Computer Animation and Virtual Worlds*, vol. 15, n° 1, p. 39–52, 2004.

Laird J. E., Newell A., Rosenbloom P. S., « SOAR : an architecture for general intelligence », *Artif. Intell.*, vol. 33, n° 1, p. 1-64, 1987.

Larsson S., Ljunglof P., « GoDiS and the dialogue move engine toolkit », *ACL'99, 37th Annual Meeting of the Association for Computational Linguistics*, 1999.

Larsson S., Traum D., « Information state and dialogue management in the TRINDI dialogue move engine toolkit », *Natural Language Engineering*, vol. 6, n° 3&4, p. 323–340, 2001.

Lazarus R. S., *Psychological stress and the coping process*, McGraw-Hill, 1966.

Leray D., Sansonnet J., « Ordinary user oriented model construction for Assisting Conversational Agents », *CHAA'06 at IEEE-WIC-ACM Conference on Intelligent Agent Technology*, 2006.

Lester J. C., Converse S. A., Kahler S. E., Barlow S. T., Stone B. A., Bhogal R. S., « The persona effect : affective impact of animated pedagogical agents », *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, Atlanta, Georgia, United States, p. 359-366, March, 1997.

Lukasiewicz J., *Elements of Mathematical Logic*, Pergamon Press, 1963.

Maes P., « Agents that reduce work and information overload », *Commun. ACM*, vol. 37, n° 7, p. 30-40, 1994.

Malatesta L., Caridakis G., Raouzaiou A., Karpouzis K., « Agent Personality Traits in Virtual Environments Based on Appraisal Theory Predictions », *Artificial and Ambient Intelligence, Language, Speech and Gesture for Expressive Characters, AISB 07*, Newcastle, UK, 2007.

Martin J., d'Alessandro C., Jacquemin C., Katz B., Max A., Pointal L., Rilliard A., « 3D Audiovisual Rendering and Real-Time Interactive Control of Expressivity in a Talking Head », *Proc. of IVA'2007*, p. 29-36, 2007.

McRorie M., Sneddon I., de Sevin E., Bevacqua E., Pelachaud C., « A model of personality and emotional traits », *Intelligent Virtual Agents (IVA 2009)*, vol. 5773 of *LNAI*, Springer-Verlag, Amsterdam, NL, p. 27-33, 2009.

Norling E., Ritter F. E., « Towards Supporting Psychologically Plausible Variability in Agent-Based Human Modelling », *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2004.

Ogborn J. M., Johnson L., « Conversation theory », *Kybernetes*, vol. 13, n° 1, p. 7-16, 1984.

Ortony A., Clore G. L., Collins A., *The Cognitive Structure of Emotions*, cambridge university press edn, Cambridge, UK, 1988.

Pelachaud C., « Some considerations about embodied agents », *Proc. of the Workshop on "Achieving Human-Like Behavior in Interactive Animated Agents", in The Fourth International Conference on Autonomous Agents*, Barcelona, Spain, 2000.

Quesada J. F., Torree D., Amores J. G., *Design of a Natural Command Language Dialogue System*, SIRIDUS IST-1999-10516, 2000.

Rao A. S., Georgeff M. P., « Modelling Rational Agents within a BDI Architecture », *in* R. Fikes, E. Sandewall (eds), *Proceedings of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, CA, USA, p. 473-484, 1991.

Rao A. S., Georgeff M. P., « BDI Agents : From Theory to Practice », *Proc. First Int. Conference on Multi-agent Systems (ICMAS-95)*, p. 312-319, 1995.

Reiter R., « An experimentation with novice users. », *H. Gallaire and J. Minker, editors, Logic and Data Bases*p. 119-40, 1978.

Rich C., Sidner C. L., « COLLAGEN : When Agents Collaborate with People », *First International Conference on Autonomous Agents*, Marina del Rey, CA, p. 284-291, 1997.

Rich C., Sidner C. L., Lesh N., « COLLAGEN : Applying collaborative discourse theory to human/computer », *AI Magazine, Special Issue on Intelligent User Interfaces*, vol. 22, n° 4, p. 15-25, 2001.

Rizzo P., Veloso M. V., Miceli M., Cesta A., « Goal-based personalities and Social behaviors in Believable Agents », *Applied Artificial Intelligence*, vol. 13, n° 3, p. 239 - 271, 1999.

Rousseau D., Hayes-Roth B., « Personality in synthetic characters », *Technical report, KSL 96-21*, Knowledge Systems Laboratory, Stanford University, 1996.

Russell S., Norvig P., *Artificial Intelligence : A Modern Approach*, 3rd edn, Prentice Hall Press, 2009.

Sabater J., Sierra C., « Review on Computational Trust and Reputation Models », *Artificial Intelligence Review*, vol. 24, n° 1, p. 33-60, 2003.

Sansonnet J. P., Description scientifique du projet Interviews - The Interviews project, Technical Report n° 99-01, LIMSI-CNRS, Orsay, France, 1999.

Sansonnet J. P., « Composants dialogiques generiques. Perspectives et methodes pour une approche integree », *Revue RSTI l'objet*, vol. 102, n° 3, p. 189-202, 2004.

Sansonnet J. P., « DIVA - DOM Integrated Virtual Agent », , `http://www.limsi.fr/~jps/online/diva/divahome/index.html`, 2008.

Sansonnet J. P., Bouchet F., « Extraction of Agent Psychological Behaviors from Glosses of WordNet Personality Adjectives », *Paper submitted at the Proc. of the 8th European Workshop on Multi-Agent Systems (EUMAS'10)*, Paris, France, 2010.

Sansonnet J. P., Leray D., Martin J. C., « Architecture of a Framework for Generic Assisting Conversational Agents », *International Conference on Intelligent Virtual Agents IVA'06*, vol. 4133 of *LNAI*, Springer, CA : Marina del Rey, p. 145-156, 2006.

Scherer K. R., Schorr A., Johnstone T., *Appraisal processes in emotion : Theory, methods, research*, Oxford University Press, 2001.

Silverman B. G., Cornwell M., O'Brien K., « Human Behavior Models for Agents in Simulators and Games : Part I : Enabling Science with PMFserv », *PRESENCE : Teleoperators and Virtual Environments*, vol. 15, p. 139-162, 2006.

Wooldridge M., *Reasoning about Rational Agents*, MIT Press, July, 2000.

Wooldridge M., Jennings N. R., « Intelligent agents : Theory and practice », *The Knowledge Engineering Review*, vol. 10, n° 2, p. 115-152, 1995.

Wu Y.-L., Tao Y.-H., Yang P.-C., « The Discussion on Influence of Website Usability Towards User Acceptability », *International Conf. on Management and Service Science (MASS'09).*, p. 1-4, sept., 2009.

Xuetao M., Bouchet F., Sansonnet J. P., « Impact of agent's answers variability on its believability and human-likeness and consequent chatbot improvements », *in* G. Michaelson, R. Aylett (eds), *Proc. of the Symposium Killer Robots vs Friendly Fridges – The Social Understanding of Artificial Intelligence (AISB 2009)*, SSAISB, Edinburgh, Scotland, p. 31-36, April, 2009.