Apports Complémentaires de la Subjectivité et des Biais Cognitifs à la Rationalité dans le Contexte de la Fonction d'Assistance

François Bouchet⋆ Jean-Paul Sansonnet⋆
bouchet@limsi.fr jps@limsi.fr

⋆ LIMSI-CNRS
Université Paris-Sud XI
BP 133, F-91403 Orsay Cedex

Résumé :
Les agents conversationnels sont un moyen prometteur d'assister des utilisateurs novices. Après une analyse sémantique, les requêtes en langue naturelle sont transformées en une représentation formelle utilisée en conjonction avec le modèle de l'application pour définir la réaction la plus appropriée. Cependant, les heuristiques associant des comportements à des schémas de requêtes sémantiquement similaires ne parviennent souvent pas à fournir une réaction efficace et réaliste quand elles se basent uniquement sur des décisions purement rationnelles. Nous proposons donc dans cet article une architecture d'agents conversationnels assistants fondée sur deux éléments : des heuristiques prenant en compte des paramètres à la fois rationnels et subjectifs (basés sur un modèle de personnalité de l'agent), et des biais utilisés pour modéliser des contraintes profondément liées à sa personnalité que l'agent ne peut modifier. Nous illustrons son fonctionnement sur des requêtes typiques issues d'un corpus de requêtes collectées avec un agent assistant.

Mots-clés : Agent conversationel, assistance, heuristiques, personnalité, biais cognitif

Abstract:
Conversational agents are a promising way to provide assistance to novice users. After a semantic analysis, natural language requests are transformed into a formal representation the agent is using in conjunction with a model of the application to define the most appropriated reaction. However, in many cases, heuristics associating behaviors to patterns of semantically similar requests fail to provide a reaction both efficient and realistic when they are only based on purely rational decisions. To face this issue, we propose in this article an architecture for assisting conversational agents based on two elements : heuristics taking into account both rational and subjective parameters (based on a psychological model of the agent), and biases used to model deep personality contraints that the agent is unable to modify. We illustrate its functioning over some typical requests extracted from a collected corpus of requests to an assisting agent.

Keywords: Conversational agent, assistance, heuristics, personality, cognitive bias

1 Introduction

1.1 Contexte

Les chercheurs ont montré que quand des utilisateurs novices placés face à un logiciel inconnu ont besoin d'assistance, ils ont tendance à demander de l'aide à "un ami derrière l'épaule" plutôt que d'avoir recours au système d'aide traditionnel disponible sur leur ordinateur [3]. Si ce phénomène s'explique par la saillance cognitive de la tâche en cours, appelé "paradoxe de la motivation" [4], il n'est pas interdit de penser qu'il est aussi lié au besoin d'une interaction plus intuitive et plus compréhensive. Cette dernière hypothèse est confirmée par l'observation du 'Persona Effect' par Lester [12], ainsi que de l'effet positif de l'interaction en Langue Naturelle dans le contexte de la Fonction d'Assistance. Tout ceci suggère que l'emploi d'Agents Conversationnels Animés (ECA – pour Embodied Conversational Agents [6]) pour aider les utilisateurs, en particulier les novices, pourrait se révéler efficace. Dans cet article, nous nous intéressons tout particulièrement à la sous-classe d'ECAs dédiés à la Fonction d'Assistance, que nous appellerons donc "Agents Conversationnels Assistants" (ACA).

Dans le cas où l'on considère les agents assistants munis d'une personnalisation (*i.e.* d'une apparence physique sous forme d'un personnage animé à l'écran), plusieurs études ont été menées afin d'améliorer la crédibilité physique des agents, en particulier via l'expression des émotions de base [1][13]. Dans cette veine, nous pensons que pour franchir la "vallée dérangeante" [14] qui fait obstacle à l'acceptabilité de ces outils par le grand public, nous aurons besoin dans le futur d'agents qui ne seront pas seulement *physiquement crédibles* mais aussi *cognitivement crédibles*, c'est-à-dire capables de produire des comportements complexes, proches de ceux des êtres humains. Une façon d'aller dans cette direction est de munir les agents : d'une part, de paramètres de personnalité semblables à ceux utilisés dans les études en psychologie humaine, et d'autre part, de contraintes cognitives intégrées au cœur même de l'architecture supportant les agents afin d'émuler les restrictions que des humains auraient en des circonstances similaires.

1.2 Travaux connexes

Dans le domaine de la simulation des communautés humaines à base de Systèmes Multi-Agents, la notion "d'agents cognitifs" (c'est-à-dire utilisant des théories cognitives afin de modéliser les capacités de raisonnement rationnel des agents) a déjà été explorée, par exemple via l'ajout d'une couche de traitement au-dessus des plateformes de déploiement d'agent traditionnelles comme, Co-JACK [15][8] pour JACK qui prend en compte les paramètres de simulation de certaines contraintes psychologiques humaines comme : les délais cognitifs, la limite de la mémoire de travail (par exemple, "oublier une croyance" si l'activation est faible ou encore "oublier quoi faire ensuite" dans une procédure donnée), l'utilisation du flou pour la remémoration des croyances, un domaine attentionnel limité, ou encore l'utilisation de 'modérateurs' altérant la cognition. Il y a eu aussi des tentatives d'ajout d'émotions aux architectures classiques de type BDI [18], par exemple pour prendre en compte la peur, l'anxiété ou la confiance en soi par l'ajout de paramètres comme les désirs fondamentaux, les capacités et les ressources [17]. L'idée d'ajouter des degrés en logique multivaluée pour les croyances, les désirs et les intentions a été étudiée dans [5] au moyen de la logique de Łukasiewicz. Ceci a conduit à montrer que l'ordre dans lequel les heuristiques sont appliquées a un impact essentiel sur la personnalité de l'agent, telle qu'elle est perçue par les humains : si nous considérons des classes de règles (comme les BDI ou encore les Obligations), ce peut aussi être une manière de caractériser la personnalité de l'agent avec des traits comme Stable, Égoïste ou Social [7].

1.3 Le besoin de personnalité pour un ACA

Dans cet article, nous mettons l'accent sur cette question essentielle dans le contexte des ACAs en considérant des situations où le raisonnement rationnel de l'agent seul ne suffit pas à produire une réaction pertinente en termes de :

– *Compétence* : la capacité de produire non seulement une réponse utile pour une requête utilisateur donnée mais aussi une réponse qui satisfasse les intentions de l'utilisateur ? celles-ci n'étant pas forcément exprimées de manière explicite dans sa requête mais pouvant être liées à la pragmatique linguistique (ex : "Dommage qu'on ne puisse revenir à la page d'accueil" → "Revenir à l'accueil").
– *Réalisme* : la capacité à réagir de manière similaire à un assistant humain placé dans la même situation et qui plus est en maintenant un comportement consistant (par exemple ne pas basculer brutalement, sans raison, d'un comportement coopératif à un comportement antagoniste). Dans l'étude des émotions, la consistance a été identifiée comme un facteur crucial par Ortony [16].

Considérons par exemple le cas d'un agent recevant une requête simple, où l'utilisateur demande comment faire pour quitter l'application "Comment je fais pour quitter ?". D'un point de vue purement rationnel, l'agent devrait répondre par exemple en désignant d'un geste déictique le bouton 'QUIT' ou encore en détaillant textuellement une procédure à suivre. En termes de compétence, ces réponses sont valables car elles satisfont parfaitement l'objectif explicitement exprimé par l'utilisateur. En termes de réalisme, cette interprétation n'est pas suffisante car elle ne prend pas en compte l'implication que porte cette simple demande qui n'est pas neutre psychologiquement. En fait, si on considère un assistant humain, selon le contexte d'interaction et selon sa personnalité, il pourrait avoir les réactions suivantes :

– *Surprise* : par exemple la tâche en cours n'est pas achevée (sensibilité au contexte) ;
– *Déception* : si par exemple il apprécie l'utilisateur (sensibilité à la subjectivité) ;
– *Satisfaction* : si au contraire l'utilisateur a été impoli (sensibilité à la subjectivité).

Ainsi, même si il y a des situations dans lesquelles un comportement purement rationnel est acceptable (dans une situation de contrôle/commande stricte ou encore dans le cas d'un agent ayant des traits de personnalité comme Servile ou Introverti), la décision doit tout de même être motivée sur la base des interactions précédentes et de la personnalité de l'agent. Si au contraire on se fonde sur des réactions purement rationnelles, le comportement de l'agent risque de n'être pas compris/admis par l'utilisateur, en raison de la tendance naturelle des humains à l'anthropomorphisme consistant à attribuer des traits psychologiques à des objets technologiques [19].

1.4 Le besoin de contraintes cognitives pour un ACA

Un autre aspect du réalisme concerne la façon dont l'agent prend les décisions mentionnées ci-dessus. En effet, dans un système à base d'heuristiques :

1) même les décisions conduisant à exprimer sa personnalité sont toujours calculées intentionnellement ; on entend par là que l'agent doit être capable de produire, si on le lui demande, les règles qui ont été appliquées et donc déterminer son comportement rationnel de base sans elles ;
2) si nous considérons un agent dans lequel les heuristiques sont construites par des concepteurs d'agents différents et indépendants, chaque règle peut être potentiellement cachée ou altérée par d'autres règles de priorité supérieure (*i.e.* appliquées après). Ceci peut poser problème dans le cas où des comportements fortement émotionnels sont produits. Par exemple, si nous considérons que l'agent est rendu 'en colère', celle-ci pourrait ne pas etre exprimée à cause d'une autre règle stipulant que l'agent ne devrait pas exprimer de comportements extrêmes (socialement non convenables) – un 'self-control' que n'a pas toujours un être humain soumis à de fortes émotions ;
3) si nous considérons le cas d'un agent qui apprend un comportement optimal en fonction de ses interactions avec des utilisateurs humains (ce que Sloman nomme 'self-monitoring' dans [21]), il pourrait finir par se débarrasser de certaines heuristiques qui augmentent son réalisme mais qui diminuent sa compétence. En effet, alors que l'efficacité peut se mesurer en comparant les résultats prédits par les heuristiques à l'état du monde réel obtenu en les appliquant ou d'après les retours langagiers des utilisateurs, il semble difficile d'imaginer une manière pour un agent d'évaluer son réalisme.

Pour toutes ces raisons, il faut implémenter les contraintes cognitives de manière suffisamment profonde pour rendre l'agent incapable a) d'expliquer son comportement et b) de les modifier. Certes cette approche est contre-intuitive vis-à-vis des systèmes à base de règles traditionnels, mais c'est le point de vue que nous allons adopter dans cet article, en introduisant la notion de "biais cognitifs" (dans la suite, simplement *biais*) qui sont des règles de transformation classiques mais avec deux différences essentielles les rendant assimilables à des filtres :
a) elles agissent comme des règles cachées : comme elles sont appliquées en-dehors du moteur de règles principal de l'agent, leur impact n'est pas analysable par l'agent ;
b) elles agissent sur les interactions même de l'agent d'une part avec le Monde (extérieur) et d'autre part avec sa propre Mémoire (intérieur).
L'implémentation garantit que ces règles/filtres seront toujours les premiers/derniers à être appliqués ce qui les rend non-altérables par les heuristiques définissables par les concepteurs d'agents assistants. Enfin elles "fonctionnent à perte" ce qui revient à dire que le fait de les appliquer efface la requête utilisateur originelle qui n'est plus désormais accessible à l'agent. Par exemple dans l'agent mécontent mentionné plus haut, il ne pourrait plus contrôler l'expression de sa colère dans ses réponses ; en fait, il n'en serait même pas conscient (*i.e.* pas capable d'en posséder une représentation symbolique dans sa base de connaissances).

Afin d'exhiber ce que pourraient être les paramètres de personnalité pertinents pour un agent assistant, il est indispensable de procéder à des expérimentations avec diverses classes d'agents. Ceci débouche sur la nécessité de développer une architecture capable de supporter cette variabilité. Dans cet article, nous commençons par la proposition d'une architecture d'agent rationnel dans laquelle les heuristiques sont capables d'entremêler subjectivité et rationalité afin d'exprimer des comportements réalistes, en particulier par le recours à un modèle psychologique de la personnalité de l'agent. Dans une seconde étape, nous montrons comment il est possible de modifier cette architecture pour modéliser la notion de biais cognitif au travers d'exemples de contraintes cognitives que les humains peuvent faire apparaître dans leurs réactions. Enfin, nous concluons par une discussion au sujet de notre approche dans le contexte des agents conversationnels dédiés à la Fonction d'Assistance.

2 Architecture d'agent Rationnel et Subjectif

2.1 Notations du modèle

Nous définissons une *entité* comme étant toute notion que nous voulons réifier (*i.e.* représenter symboliquement). Formellement une entité est représentée par un ensemble de triplets (à la manière de RDF [10]) associée à un identifiant de la forme :

$$\#id = H\left[\bigcup_i a_i \rightarrow v_i\right]$$

Où $\#id$ dénote l'*identifiant* unique donné arbitrairement à la référence, $H \in \mathbb{H}$ est la *tête* de l'entité, a_i un attribut pris dans la liste des attributs disponibles pour H et v_i une valeur prise dans le domaine des expressions définies par le type associé à a_i. En fonction du type, v_i peut être :

– une valeur terminale (chaîne de caractères, nombre, valeur symbolique énumérée...),
– une nouvelle entité, selon ce même format d'entité ;
– un identifiant correspondant à une entité déjà définie.

Nous définissons aussi un *moteur* ou *engine* comme étant une façon de générer une nouvelle entité ou bien encore de transformer une entité en une autre, grâce à un ensemble de règles définissant le moteur.

Nous appelons *domaine* un ensemble d'éléments (entités ou moteurs) qui ont un accès direct les uns aux autres. Ainsi, nous distinguons quatre domaines différents :

1. *La Mémoire de l'agent* ($\mathcal{M}$), qui stocke toutes les connaissances que l'agent possède à l'origine ou acquises lors de ses interactions. Plus particulièrement, la mémoire se décompose en trois parties :
   – *La mémoire sémantique* ($\mathcal{M}_s$), qui contient le modèle partiel du Monde que l'agent a pu acquérir de manière directe par consultation ou indirecte par inférence.
   – *La mémoire épisodique* ($\mathcal{M}_e$), dont la différence avec $\mathcal{M}_s$ vient du fait qu'elle se focalise sur l'agent lui-même, représentant ainsi sa mémoire autobiographique (*i.e.* "ce qui lui est arrivé", *cf.* [22]). Dans le cas d'agents conversationnels, l'agent ne connait le Monde qu'au travers de ses interactions avec l'utilisateur d'une part et l'application assistée d'autre part : sa mémoire épisodique est alors réduite à l'enregistrement des entrées/sorties interactionnelles [1].
   – *La mémoire procédurale* ($\mathcal{M}_p$), qui est composée d'un ensemble d'heuristiques, c'est-à-dire de règles à appliquer selon certaines conditions. Pour cette étude, nous supposerons qu'elles sont déclenchées uniquement lors de la réception d'une requête de l'utilisateur.
2. *Les états mentaux de l'agent* (Ψ) contiennent l'information concernant la psychologie de l'agent, modélisée selon quatre types de notions : traits (traits), humeurs (moods), rôles (roles) et affects (relationships).
3. *Le moteur de l'agent*, ou encore *engine* ($\mathcal{E}$), chargé d'exécuter la partie active de l'agent. Il se décompose en :
   – Un moteur de Traitement Automatique de la Langue Naturelle ($\mathcal{E}_L$), qui transforme les questions en langue naturelle des utilisateurs en une représentation formelle de leur sémantique.
   – Un moteur comportemental ou encore *behavioral* ($\mathcal{E}_B$), qui détermine quelles heuristiques (dans $\mathcal{M}_p$) doivent être activées en fonction de la requête [2] et qui calcule les réactions en fonction de ces heuristiques en les paramétrant par les valeurs actuelles des variables y apparaissant.
4. Le Monde ($\mathcal{W}$) est défini, du point de vue de l'agent comme étant tout ce qui n'est pas interne (*i.e.* ni dans $\mathcal{E}$, Ψ ou $\mathcal{M}$). En particulier, $\mathcal{W}$contient les informations sur les utilisateurs et l'application assistée. Nous supposerons que le Monde existe et évolue de manière indépendante de l'agent, même si l'agent peut influer sur le Monde.

2.2 Représentation du Monde ($\mathcal{W}$)

Le Monde est constitué d'entités représentées dans la syntaxe définie au § 2.1. Par exemple, l'information disponible au sujet d'un utilisateur donné peut être représentée comme :

```
#user7 = PERSON[
            name   -> "Smith",
            role   -> user,
            age    -> 20,
            gender -> male
]
```

2.3 États mentaux de l'agent (Ψ)

Nous distinguons quatre types d'états mentaux en fonction de leur dynamique et de leur arité fonctionnelle, tels que résumés dans la table 1. À chacun d'entre eux est associée une valeur numérique dans l'intervalle réel $[-1, 1]$, 0 définissant par défaut "l'état mental neutre". Les états mentaux sont représentés sous forme attribut-valeur de l'agent comme par exemple :

```
#ums = unary-mentalstate[
   mentalstate1 -> 0.8,
   mentalstate2 -> -0.3,
   ...
]
#bms = binary-mentalstate[
```

1. Bien que les interactions entre l'utilisateur et l'agent soient en théorie de nature multimodale, nous ne considérerons ici que la modalité linguistique comme support des interactions utilisateur/agent.

2. Nous considérerons dans cet article que l'agent est de type purement réactif (ou modal) : il ne prend pas l'initiative de démarrer une interaction avec l'utilisateur.

TABLE 1 – Les quatre types d'états mentaux d'un agent, selon leur dynamique et leur arité

	Unaire	**Binaire**
Statique	Trait Ψ_T	Role Ψ_R
Dynamique	Humeur Ψ_t	Affect Ψ_r

```
   towards -> #iduser,
   mentalstate1 -> 1,
   mentalstate2 -> 0,
   ...
]
```

Traits Ψ_T. Les traits correspondent aux attributs classiques de la personnalité qui peuvent être considérés comme stable au cours de la 'vie' d'un agent – ils correspondent aux "Big Five" ou "modèle OCEAN", couramment utilisé en psychologie [9] :

- *Ouverture à l'expérience* : représente l'imagination, la curiosité, le goût pour l'aventure ;
- *Caractère Consciencieux* : représente la tendance à être discipliné et à respecter ses obligations ;
- *Extraversion* : représente l'énergie, la force des émotions positives et le goût des autres ;
- *Agréabilité* : représente la tendance à être compatissant, affectueux et coopératif ;
- *Neuroticisme* : représente la tendance à ressentir facilement des émotions négatives (colère, anxiété, vulnérabilité *etc.*).

Humeurs (Moods) Ψ_t. Les humeurs (ou moods) représentent les facteurs de personnalité qui varient avec le temps, en fonction des heuristiques et des biais cognitifs. L'état humoral d'un agent est divisé en deux parties :

1. Propriétés physiques : elles modélisent l'agent en tant qu'entité physique du monde (à la manière de la métaphore des attributs physiques des jeux vidéo). Nous ne les traiterons pas dans cet article ;
2. Propriétés épistémiques : elles modélisent l'agent en tant qu'entité capable de raisonnement rationnel. C'est pourquoi elles seront prises en compte dans cette étude.

En résumé, nous aurons les humeurs suivantes :

- *Energy (énergie physique)* : représente la force physique de l'agent, au sens large ;
- *Happiness (bonheur physiologique)* : représente le bien-être physique de l'agent, selon sa situation physique ;
- *Confidence (confiance en soi)* : représente la force cognitive de l'agent ;
- *Satisfaction (satisfaction intellectuelle)* : représente le bien-être mental de l'agent, selon l'analyse qu'il fait de sa situation intentionnelle (*i.e.* vis-à-vis de ses B-D-I).

Rôles Ψ_R. Les rôles représentent des relations statiques à caractère institutionnel, entre l'agent et d'autres entités du monde (*e.g.* la relation utilisateur/assistant). On peut définir deux grandes catégories :

- *Authority (autorité)* : elle caractérise le droit que l'agent croit posséder d'être directif vis-à-vis de son interlocuteur ou réciproquement de ne pas accepter (facilement) des directives de celui-ci. Cette relation est souvent antisymétrique : `Authority(X,Y) = -Authority(Y,X)`
- *Familiarity (familiarité)* : elle caractérise le droit que l'agent croit posséder de se comporter de manière informelle vis-à-vis de son interlocuteur. Cette relation est souvent symétrique : `Familiarity(X,Y) = Familiarity(Y,X)`

Dans le cas de la Fonction d'Assistance, l'autorité sera a priori clairement en faveur de l'utilisateur et nous aurons par conséquent :

```
role[
   towards -> #iduser,
   authority -> val1,
   familiarity -> val2
]
```

dans lequel 'val1' doit avoir une valeur négative.

Affects (relationships) Ψ_r. Les affects modélisent les relations dynamiques entre l'agent et les autres entités (typiquement les utilisateurs). Nous en distinguons au moins trois sortes :

- *Dominance* : exprime que l'agent se sent puissant par rapport à l'interlocuteur. Cette relation est souvent antisymétrique, comme par exemple `Dominance(X,Y) = -Dominance(Y,X)` ;
- *Affection* : exprime une attirance et une tendance à agir amicalement de l'agent envers l'interlocuteur. Cette relation n'est pas nécessairement symétrique ;
- *Trust (confiance)* : exprime que l'agent fait confiance à l'interlocuteur. Cette relation n'est pas nécessairement symétrique.

2.4 La mémoire de l'agent ($\mathcal{M}$)

Mémoire épisodique $\mathcal{M}_e$. Elle contient les interactions précédentes de l'agent avec d'une part l'utilisateur et d'autre part l'application. Nous distinguons les messages entrants (`INBOX`) des messages sortants (`OUTBOX`). L'information est représentée par des triplets de la forme :

```
INBOX[
   from -> [sender],
   time -> [timestamp],
   message -> [message]
]
OUTBOX[
   to -> [recipient],
   time -> [timestamp],
   message -> [message]
]
```

Mémoire sémantique $\mathcal{M}_s$. Elle contient un sous-ensemble étendu du Monde et de ce fait utilise la même représentation. Il s'agit d'un sous-ensemble car la totalité de $\mathcal{W}$ est souvent inatteignable pour l'agent ; il s'agit d'une extension car l'agent crée à tout moment de nouveaux faits lors de l'exécution des heuristiques et des filtres de biais. La manière dont l'agent construit cette mémoire de manière automatique grâce à des heuristiques d'observation du Monde (ou *observateurs*) sort du cadre de cet article, mais des éléments sont donnés dans [11].

Mémoire procédurale $\mathcal{M}_p$. Elle contient un ensemble d'heuristiques définissant comment l'agent doit réagir à une requête entrante. Une heuristique contient deux parties :

- une *tête*, définissant les classes de requêtes avec lesquelles les heuristiques doivent s'apparier selon la syntaxe des expressions régulières. Par exemple, les requêtes concernant la possibilité d'exécuter une action.
- un *corps*, définissant un arbre de décision permettant de construire les réactions de l'agent à cette requête, chaque nœud de l'arbre étant basé sur une combinaison de valeurs retournées par les requêtes envoyées vers $\mathcal{M}$ et $\mathcal{W}$ (pour la partie de la réponse liée à la rationalité) ou bien vers $\mathcal{M}_s$ (pour la partie de la réponse liée à la subjectivité). Cela se termine toujours par un envoi de requête de performatif de type `INFORM` vers le Monde.

Exemple : Réaction d'un agent quand l'utilisateur lui interdit d'effectuer une action donnée, par exemple : "Je te défends d'ouvrir le fichier de configuration !". Il s'agit d'une réaction très subjective définie par le code suivant :

```
HEURISTIC[
   description -> "reaction to an interdiction",
   head -> NEG[AUTHORIZATION[
                granter -> person[id="user"],
                granted -> person[id="system"],
                todo -> A___]],
   body -> {
# If the agent is conscientious, it first checks
# if the action was doable in first place
if (conscientiousness > 0):
 # Choice of the repository to check: world or memory
 repository = (confidence > 0.5 ? W ; MS)
 # Check if the action is doable at first
 allowed = CHECK[repository, DOABLE[A], true]

# If the information is not found in the memory but
# was in the world, the agent loses some confidence
 if (allowed == unknown && repository = MS):
  allowed = CHECK[W, DOABLE[A], true]
  if (allowed != unknown):
   INFORM[memory, decrease(confidence)]

# If the action was not doable,
# the agent gets more confidence in itself
 if (allowed == false):
  INFORM[memory, increase(confidence)]
  # If it's cooperative, it says it, + or - nicely
  if (agreeableness > 0):
   if (affection(user)>=0 && familiarity(user)>=0):
    answer += POSITIVE[NOTPOSSIBLE[A___]];
   elif (affection(user) < -0.5):
    answer += NEGATIVE[NOTPOSSIBLE[A___]];
   else:
    answer += NOTPOSSIBLE[A___];

# If the user has authority the fact is stored
if (authority(user) > 0):
 INFORM[memory, forbidden(A___)]
 done = true
else:
 done = false

# If the agent is neurotic,
# being forbidden something makes it unsatisfied
if (neuroticism > 0):
 INFORM[memory, decrease(satisfaction)]
 # If it has a high dominance, it will mention it
  if (dominance(user) > 0):
   answer += UNHAPPY

# The action done is acknowledge
# possibily with a negative modelization
# which expression depends on the familiarity,
if (satisfaction < -0.3 && familiarity(user) > 0):
 answer += NEGATIVE[(done?ACK:NACK)]
elif (done && satisfaction < -0.8):
 answer += NEGATIVE[(done?ACK:NACK)]
else:
 answer += (done?ACK:NACK)

# it finally transmits the built answer to the user
INFORM[user, answer]
 }
]
```

Remarques :

- La réaction est construite progressivement dans la variable 'answer' ;
- La génération effective de la Langue Naturelle n'est pas traitée ici : on se contente d'employer des patrons comme NOTPOSSIBLE, NEGATIVE ou ACK qui devraient être post-traités. Ces symboles pourraient aussi être interprétés de manière multimodale, par exemple pour choisir les animations physiques de la personnification de l'agent.

Le contenu du corps de cette heuristique sera détaillé en 2.5.

2.5 Fonctionnement dynamique

Le moteur de traitement de la Langue Naturelle $\mathcal{E}_L$. Il a pour rôle de prétraiter les requêtes en Langue Naturelle pour les mettre sous forme de représentation formelle transmise à $\mathcal{E}_B$, selon le langage défini dans [2]. Typiquement, les requêtes en Langue Naturelle sont traitées en deux étapes :

- Une phase d'analyse grammaticale : les outils de TALN classiques sont appliqués (lemmatisation, étiquetage lexical, désambigüisation sémantique...) ;
- Une phase d'analyse sémantique : une requête formelle est construite selon le langage défini dans [2].

Moteur comportemental $\mathcal{E}_B$. Le moteur comportemental de l'agent accède régulièrement à l'information stockée dans d'autres domaines ($\mathcal{M}$, Ψ ou $\mathcal{W}$). Par définition, $\mathcal{E}$ ne peut pas accéder directement aux autres domaines, il doit utiliser à cet effet des *requêtes*. Dans cet environnement simplifié, nous faisons une distinction entre trois sortes de requêtes :

- `INFORM[domain, request]` : transmet le contenu de la requête à un domaine. Cette requête n'attend pas de requête en retour.
- `GET[domain, value]` : requiert la valeur d'un élément auprès d'un domaine. Cette requête attend une requête `INFORM[X,Y]` en réponse de la part du domaine interrogé.
- `CHECK[domain, attribute, value]` : demande si la valeur d'un attribut du domaine interrogé est bien celle fournie comme troisième paramètre de la requête. Cette requête attend une requête `INFORM[X,Y]` en réponse de la part du domaine interrogé où la valeur peut être Faux, Vrai ou Inconnu.

Remarque : En raison du modèle de requêtes simplifié choisi, nous ne détaillons pas la sémantique du protocole à la manière de l'ACL-FIPA [20] qui utilise des pré-conditions et post-conditions en logique modale.

Protocole d'Interaction général. Ici, la seule façon dont l'utilisateur peut interagir avec l'agent est via l'interaction textuelle. En conséquence, une interaction commence toujours par une entrée en Langue Naturelle émanant de l'utilisateur qui déclenche la séquence d'événements suivante (*cf.* figure 1) :

1. La requête est traitée par $\mathcal{E}_L$ qui fournit une représentation formelle à $\mathcal{E}_B$;
2. La représentation formelle de la requête entrante est stockée dans $\mathcal{M}_e$;
3. En fonction de la représentation formelle de la requête et des patrons définis dans la tête de chaque heuristique stockée dans $\mathcal{M}_p$, $\mathcal{E}_B$ récupère le corps associé à l'heuristique déclenchée et exécute les règles qu'il contient. Ceci conduit à la génération d'une ou de plusieurs requêtes vers d'autres domaines :
   - si les règles requièrent l'utilisation d'éléments subjectifs, ils sont récupérés dynamiquement à la demande depuis Ψ ;
   - si les règles requièrent l'utilisation d'informations au sujet de l'application, ils sont récupérés dynamiquement à la demande depuis $\mathcal{M}_s$ ou $\mathcal{W}$;
4. La réaction formelle venant d'être construite est envoyée vers $\mathcal{E}_L$ afin de générer une réponse en Langue Naturelle qui doit être transmise à l'utilisateur.

Durant le traitement des effets des règles, plusieurs comportements distincts sont envisageables qui dépendent essentiellement des paramètres de personnalité de l'agent. Tout d'abord pour les requêtes de type CHECK, il y a deux possibilités :

1. Priorité au test dans $\mathcal{M}_s$: si l'agent a de la confiance en soi (confident) ou bien n'a plus accès à cette information à partir du Monde (par exemple, s'il s'agit d'une question relative à un état précédent de l'application) il essaye de retrouver directement l'information désirée depuis $\mathcal{M}_s$. Il ne teste $\mathcal{W}$ que s'il n'arrive pas à trouver une réponse dans sa propre mémoire.
2. Priorité au test dans $\mathcal{W}$: si l'agent n'a pas confiance en lui (not confident), il ne se donnera pas la peine de vérifier dans sa propre mémoire et essayera de retrouver l'information depuis $\mathcal{W}$.

Alors se pose le problème de savoir ce qui doit être stocké ou non dans $\mathcal{M}_s$ (stockage non représenté dans la figure 1) quand une information est récupérée depuis le Monde. À nouveau, deux options sont possibles :

1. Copie stricte de $\mathcal{W}$: pour un agent sérieux (Conscientious), la mémoire de l'agent se comporte exactement comme un cache et un historique du Monde ;
2. Copie surchargée de $\mathcal{W}$: pour un agent prêt à prendre plus de risques, au plan épistémique, la mémoire de l'agent contient aussi le résultat des calculs qu'il est amené à effectuer en interne. Par exemple, si on lui demande combien l'application contient d'objets d'un certains type (*e.g.* des boutons), il

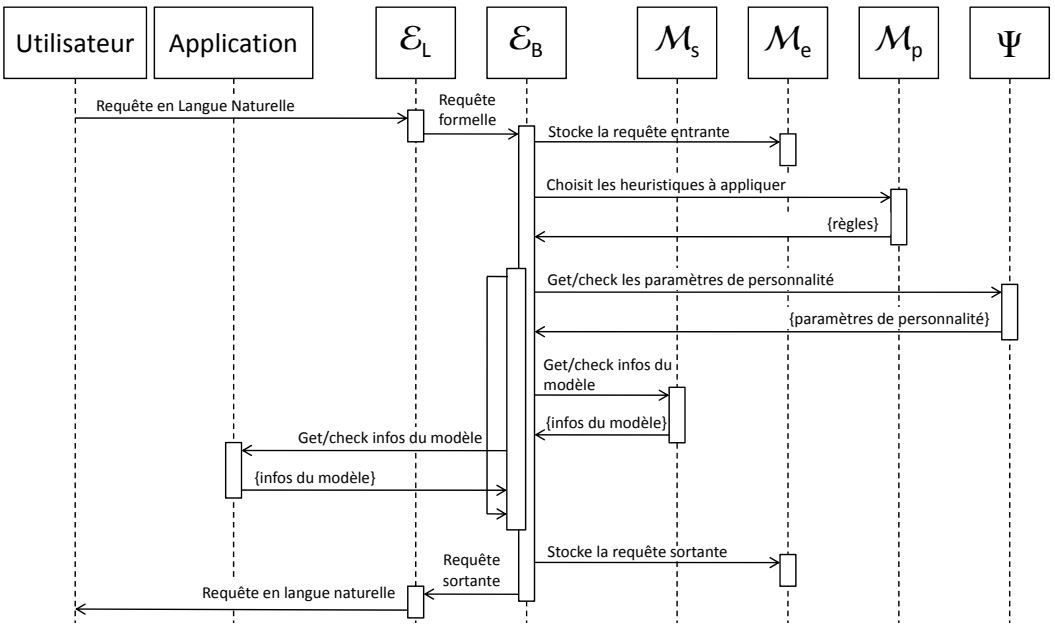

FIGURE 1 – Diagramme de séquence du traitement typique d'une requête usager par l'agent.

ne se contentera pas de stocker la liste des boutons mais il synthétisera aussi un fait décrivant explicitement cette information.

À partir d'ici, on constate qu'un agent confiant en lui mais pas très sérieux peut manquer d'efficacité car il est amené à répondre directement à l'utilisateur, sur la base d'informations qui sont peut-être obsolètes. Cependant, en termes de réalisme, ce comportement émule assez bien celui d'un être humain possédant ce trait de personnalité (Conscientious = -1) et cette humeur (Confident = 1) ; et s'il peut sembler inacceptable à un utilisateur lui-même sérieux, il n'est pas sûr qu'un utilisateur ayant le même profil blâmerait l'agent pour cela.

Exemple d'interaction. Considérons une requête de contrôle ou l'utilisateur interdit à l'agent d'ouvrir un fichier donné. Nous aurons :
Phrase de l'utilisateur : "N'ouvre pas le fichier !" ⇔ " Don't open the file !"

Requête formelle produite par l'analyseur sémantique :

```
NEG[AUTHORIZATION[
  granter -> person[id="user"],
  granted -> person[id="system"],
  todo -> Open[
    element -> object[
      properties -> {
        type -> type[val="file"]
        quantity -> quantity[val=1]
      }
]]]]
```

Si nous supposons pour simplifier qu'il n'y a qu'une seule heuristique dont la tête s'apparie avec cette requête formelle, à savoir celle décrite en 2.4, l'agent générera une réponse composée de trois sous-phrases : $[not\_possible][unhappy][ack/nack]$

- $[not\_possible]$ est généré seulement si l'agent est suffisamment sérieux (Conscientious> 0.6) pour avoir pris la peine de vérifier au préalable que le fichier est techniquement ouvrable et suffisamment coopératif (Agreeableness > 0.6) pour informer l'utilisateur à ce sujet. Cette phrase peut aussi être 'modalisée' positivement ou négativement selon l'affection et la familiarité envers l'utilisateur.
- $[unhappy]$ est généré seulement si l'agent est neurotique. On peut supposer que dans un tel cas il existe une règle stipulant qu'il n'aime pas les interdictions – et qu'il se sente suffisamment dominant envers l'utilisateur pour exprimer explicitement sa rancœur dans sa réponse.
- $[ack/nack]$ est toujours généré pour faire savoir à l'utilisateur si sa commande a été prise en compte ou pas par l'agent.

Cette heuristique illustre le fait qu'un agent peut faire plus de choses qu'il ne laisse transparaître dans sa réponse à l'utilisateur. Par exemple, s'il n'est pas coopératif (Agreeableness < 0) mais sérieux (Conscientious > 0.5) il vérifiera que l'opération demandée est techniquement effectuable, indépendamment de la décision d'en informer ou non l'utilisateur dans le cas d'une

TABLE 2 – Résumé des différences entres les règles appliquées par les biais et les heuristiques

	Heuristiques	**Biais**
Objectif	Génération d'une réaction	Modification d'une requête
Portée	Une classe de requêtes d'utilisateur	Toute requête entre deux domaines
Ressources	Requête de l'utilisateur, $\mathcal{W}$, $\mathcal{M}$ et Ψ	Ψ seulement
Introspectables	Oui	Non

impossibilité technique. Le fait d'avoir effectué cette vérification, même si elle ne transparaît pas dans l'interaction, peut avoir un impact sur les états mentaux de l'agent.
Nous voyons aussi que certains paramètres de personnalité similaires (statiques ou dynamiques) comme l'autorité et la dominance peuvent être utilisés conjointement : si l'agent n'a pas d'autorité envers l'utilisateur mais se sent dominant envers lui, il peut fournir une réponse rebelle du type "Ca m'ennuie mais ok".

3 Introduction des biais cognitifs

3.1 Définition

Nous avons vu que $\mathcal{E}_B$ communique avec $\mathcal{M}$ et $\mathcal{W}$ au moyen de requêtes. Ces requêtes peuvent être modifiées lorsqu'elles transitent d'un domaine à l'autre par des filtres appelés *biais cognitifs* ou encore *biais*. Un biais agit donc comme une transformation sur les requêtes de l'agent sans que celui-ci ne puisse en avoir connaissance. Un biais b sur une requête entre deux domaines X et Y sera représenté par : $\mathrm{X}\xrightarrow{b}\mathrm{Y}$.
La différence fondamentale avec les heuristiques stockées dans $\mathcal{M}_p$ est alors l'impossibilité pour l'agent d'expliquer les biais : dans la plupart des cas, il ne peut même pas se rendre compte (par exemple par inférence) qu'ils ont été appliqués. En outre, tandis que les heuristiques produisent des requêtes, c'est-à-dire qu'elles essayent de définir une réaction pour une situation spécifique donnée, au contraire les biais sont des contraintes s'appliquant à toute réaction. Néanmoins, les biais comme les requêtes sont affectés par les valeurs des paramètres de Ψ. La table 2 résume ces différences.

3.2 Catégories et exemples de biais

Les biais sont orientés, ce qui signifie pour une paire donnée de domaines X et Y il est possible de définir deux sortes de biais : $\mathrm{X}\xrightarrow{b}\mathrm{Y} \neq \mathrm{Y}\xrightarrow{b}\mathrm{X}$. De plus les biais sont dépendants du type de requête transmise entre deux domaines, ce qui fait que si chacun des quatre domaines définis plus haut pouvaient communiquer avec chacun des trois autres il y aurait six canaux bidirectionnels portant trois types de requêtes (INFORM, GET, CHECK) ce qui engendrerait une combinatoire de $6 \times 3 \times 2 = 36$ biais différents. Cependant, la plupart d'entre eux ne sont pas pertinents pour plusieurs raisons :

- Chaque fois que des processus sont actifs dans $\mathcal{M}$ ou Ψ, nous supposerons qu'ils n'ont pas la nécessité de communiquer avec un autre domaine, et que $\mathcal{E}$ est l'unique domaine de l'agent capable de communiquer avec le Monde : $\mathcal{E}_B$ constitue donc le centre de communication de l'agent.
- Il est difficile d'imaginer des situations où l'agent ne serait pas capable de connaître exactement son propre état mental, c'est pourquoi nous ne considérerons pas l'existence de biais entre $\mathcal{E}_B$ et Ψ. En fait, nous supposerons aussi que Ψ est directement accessible à partir des heuristiques et des biais (*cf.* l'exemple d'heuristique donné plus haut où l'on accède à la valeur du paramètre Conscientious de manière directe plutôt que par l'utilisation d'une requête de la forme `GET[mentalstates, conscientiousness]`).

Une fois ces considérations posées, il ne reste plus que sept canaux unidirectionnels parmi lesquels cinq seulement peuvent être munis de biais, comme indiqué dans la figure 2. Les cinq catégories de biais restantes sont alors :

- Biais Perceptif ($\mathcal{W}\xrightarrow{B_p}\mathcal{E}_B$) : biais sur une requête `INFORM` émanant du Monde (en fait il s'agit de l'utilisateur dans le cas d'une requête en Langue Naturelle, ou encore le reste du Monde s'il s'agit de la conséquence d'une requête de type GET envoyée précédemment).
- Biais Expressif($\mathcal{E}_B\xrightarrow{B_e}\mathcal{W}$) : biais sur une requête `INFORM` envoyée vers le Monde.
- Biais de Recherche de faits en mémoire

($\mathcal{M} \xrightarrow{B_{mr}} \mathcal{E}_B$) : biais sur une requête`INFORM` émanant de la mémoire (en réponse à une requête`GET` ou `CHECK` envoyée précédemment).
– Biais de lecture mémoire ($\mathcal{E}_B \xrightarrow{B_{ma}} \mathcal{M}$) : biais sur une requête `GET` ou `CHECK` envoyée à la mémoire.
– Biais d'écriture mémoire ($\mathcal{E}_B \xrightarrow{B_{ms}} \mathcal{M}$) : biais sur une requête `INFORM` envoyée à la mémoire.

Des exemples pour ces cinq catégories de biais sont donnés dans la section 3.4.

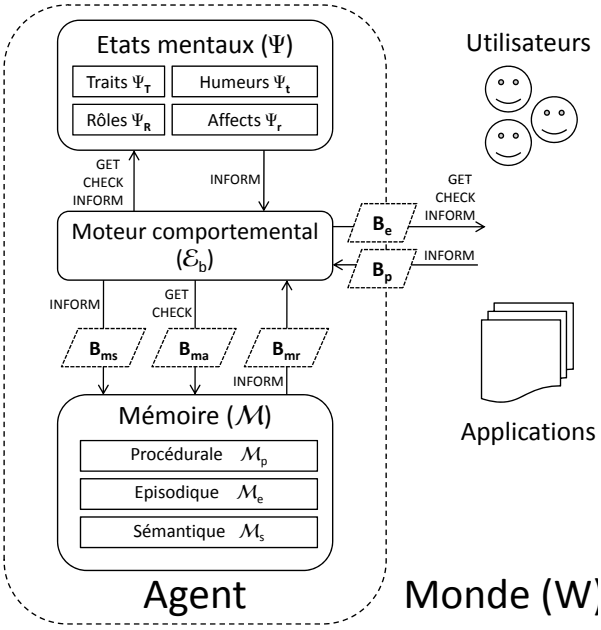

FIGURE 2 – Architecture générale de l'agent avec les canaux portant les cinq catégories de biais

3.3 Représentation des biais

Tout comme les heuristiques, les biais possèdent deux parties :
– La *catégorie* du biais : sélectionnée parmi une des cinq catégories définies ci-dessus ;
– Le *corps* : en termes formels, il n'y a pas de différence fondamentale entre la représentation du corps d'un biais et celle du corps d'une heuristique : les deux sont fondés sur un arbre de décision. Cependant, les nœuds d'un biais ne prennent en compte que les aspects de la subjectivité : ils n'ont pas accès en lecture ou en écriture aux éléments de $\mathcal{W}$ ou de $\mathcal{M}$. De plus, les actions possibles sont limitées aux modifications (directes) des paramètres dynamiques de la personnalité (humeurs et affects) ainsi que des requêtes (mais aucune requête supplémentaire ne peut être engendrée par un biais).

Par exemple, nous pouvons considérer un biais perceptif qui pourrait être appliqué par un agent nerveux (Neuroticism > 0) et malheureux (Happiness < 0) le poussant à percevoir négativement toute demande en entrée :

```
BIAS[
   description -> "victimization",
   category -> "perceptive"
   body -> {
   if (neuroticism < -0.5
       && satisfaction < -0.9):
      output = NEGATIVE[input]
   }
]
```

3.4 Exemples de biais

Biais perceptifs.
– Victimisation : comme détaillé ci-dessus, il s'agit de la tendance qu'a un agent nerveux et malheureux à percevoir plus de négativité qu'il n'y en a réellement dans une phrase de son interlocuteur ;
– Minimisation : inversement, si l'agent a un haut degré de satisfaction (confident > 0.5) et qu'il n'est pas nerveux (Neuroticism < 0), il aura tendance à sous-évaluer la charge négative contenue dans une phrase de l'interlocuteur.

Biais expressifs.
– *Stress* : si l'utilisateur est en position de forte autorité vis-à-vis de l'agent, celui-ci peut être amené à exprimer une dose de nervosité dans ses réactions. Ceci ne s'applique que pour modéliser le stress que l'agent ne peut contrôler et vient agir de manière complémentaire au "stress rationnel". En effet, il y a également du stress qui est lié au contenu propositionnel de la réponse : ce stress supplémentaire devrait être produit par une heuristique quand l'agent, par exemple, n'arrive pas à résoudre une requête posée par l'utilisateur.
– *Enjouement/tristesse* : si l'agent est expressif, il aura tendance à révéler son niveau de satisfaction épistémique par l'ajout de connotations positives (resp. négatives) à ses réponses.

Biais de recherche de faits en mémoire.
– *Doute* : si le niveau de confiance de l'agent envers ses propres connaissances est bas et que de plus il a un niveau de satisfaction bas, il peut être amené à rejeter ou tout du moins à

minimiser la valeur de certaines informations retrouvées dans sa mémoire.

Biais de lecture mémoire.
- *Mauvaise foi* : si l'agent est très insatisfait et que l'utilisateur a une forte autorité sur lui ou encore que l'agent est très mécontent, il peut être conduit à introduire de fausses informations dans les requêtes qu'il envoie vers $\mathcal{M}_s$, par exemple, par l'oubli d'un paramètre (recherche de tous les boutons alors que l'utilisateur a demandé combien il y avait de boutons rouges). L'information récupérée peut alors être partiellement fausse, mais l'agent sera "réellement convaincu d'avoir agir loyalement", et sa réponse pourrait donc presque être assimilée à un "acte manqué".

Biais d'écriture mémoire.
- *Oubli* : dans le cas ou l'agent n'est pas névrotique et qu'il est présentement satisfait, il peut choisir de ne pas enregistrer dans sa mémoire certaines informations négatives (comme par exemple une critique émise par l'interlocuteur), qu'il oublie tout simplement.
- *Désordonné* : dans le cas où l'agent n'est pas très sérieux, il peut perdre au hasard des morceaux d'information appartenant au contenu propositionnel de la requête vers $\mathcal{M}_e$sur laquelle agit le biais.

4 Conclusion

Nous avons vu que l'utilisation d'une architecture où les décisions dépendent à la fois de paramètres subjectifs et objectifs fait que l'efficacité de l'aide apportée pau un ACA devient alors fortement dépendante de ses traits de personnalité. Dans la mesure où les heuristiques prennent en compte l'état mental (dynamique) mais aussi la personnalité (statique) de l'agent, les ACAs conçus de cette manière offrent une certaine généricité et peuvent donc être adaptés :
- statiquement selon la personnalité de l'utilisateur, en choisissant des agents aux traits de personnalité similaires (Ψ_T) qui sont en général préférés par les interlocuteurs [19] ;
- dynamiquement selon les feedbacks précédents de l'utilisateur. Évidemment, comme les requêtes précédentes de l'utilisateur ont modifié les états mentaux (Ψ_t et Ψ_r) de l'agent, cela aura un impact sur ses réactions futures.

L'implémentation de biais cognitifs, indépendamment des autres règles présentes dans le corps des heuristiques, permet d'émuler certaines contraintes des comportements humains et de donner par là même une certaine primauté aux états mentaux des agents par rapport aux processus de raisonnement strictement rationnel.
L'architecture actuelle présentée dans cet article a été implémentée et interfacée avec la chaîne de traitement de requêtes en langue naturelle (sous Mathematica). Toutefois, l'impact effectif de cette approche, en particulier dans le contexte des agents assistants, reste à évaluer dans des travaux futurs qui feront intervenir des usagers novices placés face à trois classes d'agents :
- S1. Un agent purement rationnel ;
- S2. Un agent rationnel et subjectif, implémenté au moyen de l'architecture définie dans la section 2 ;
- S3. Un agent rationnel et subjectif, incluant également les biais introduits dans la section 3.

Nous nous attendons à une amélioration, en termes de réalisme, en passant de S1 à S2 ainsi que de S2 à S3. Il pourrait aussi y avoir un léger accroissement de la compétence entre S1 et S2. Cependant il est probable que l'introduction des biais conduise à une décroissance du degré de compétence perçu, ce qui pose la question du choix difficile entre compétence et réalisme.

Références

[1] Joseph Bates. The role of emotion in believable agents. *Commun. ACM*, 37(7) :122–125, July 1994.

[2] François Bouchet and Jean-Paul Sansonnet. Caractérisation de requêtes d'assistance à partir de corpus. In *Actes de MFI'07*, Paris, France, May 2007.

[3] Antonio Capobianco and Noëlle Carbonell. Contextual online help : elicitation of human experts' strategies. In *Proceedings of HCI'01*, pages 824–828, New Orleans, August 2001.

[4] John M. Carroll and Mary Beth Rosson. *Paradox of the active user*, pages 80–111. MIT Press, 1987.

[5] Ana Casali, Lluís Godo, and Carles Sierra. Graded BDI models for agent architectures. In *Proceedings of CLIMA-V*, volume 3487 of *Lecture Notes in Computer Science*, pages 126–143, Lisbon, Portugal, 2004.

[6] Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill, editors. *Em-*

bodied Conversational Agents. MIT Press, April 2000.

[7] Mehdi Dastani. A classification of cognitive agents. In *Proceedings of Cogsci02*, pages 256—261, 2002.

[8] Rick Evertsz, Franck E. Ritter, Paolo Busetta, and Matteo Pedrotti. Realistic behaviour variation in a BDI-based cognitive architecture. Melbourne, Australia, 2008.

[9] Lewis R. Goldberg. Language and individual differences : The search for universal in personality lexicons. *Review of personality and social psychology*, 2 :141–165, 1981.

[10] Ora Lassila and Ralph R. Swick. Resource description framework (RDF) model and syntax specification. W3C recommendation, 1999.

[11] David Leray and Jean-Paul Sansonnet. Ordinary user oriented model construction for assisting conversational agents. In *CHAA'06 at IEEE-WIC-ACM Conference on Intelligent Agent Technology*, 2006.

[12] James C. Lester, Sharolyn A. Converse, Susan E. Kahler, S. Todd Barlow, Brian A. Stone, and Ravinder S. Bhogal. The persona effect : affective impact of animated pedagogical agents. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 359–366, Atlanta, Georgia, United States, March 1997. ACM.

[13] Jean-Claude Martin, Christophe d'Alessandro, Christian Jacquemin, Brian Katz, Aurélien Max, Laurent Pointal, and Albert Rilliard. 3D audiovisual rendering and Real-Time interactive control of expressivity in a talking head. In *Proc. of IVA'2007*, pages 29–36, 2007.

[14] Masahiro Mori. Bukimi no tani [The uncanny valley]. *Energy*, 7(4) :33–35, 1970.

[15] Emma Norling and Franck E. Ritter. Towards supporting psychologically plausible variability in Agent-Based human modelling. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2004.

[16] Andrew Ortony. On making believable emotional agents believable. In R. Trappl and P. Petta, editors, *Emotions in humans and artifacts*. MIT Press, Cambridge, MA, 2003.

[17] David Pereira, Eugenio Oliveira, and Nelma Moreira. Formal modelling of emotions in BDI agents. In F. Sadri and K. Satoh, editors, *Proceedings of CLIMA-VIII*, volume 5056 of *LNAI*, pages 62–81, Porto, Portugal, 2008. Springer-Verlag.

[18] Anand S. Rao and Michael P. Georgeff. Modelling rational agents within a BDI architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of Knowledge Representation and Reasoning*, pages 473–484, San Mateo, CA, USA, 1991. Morgan Kaufmann.

[19] Byron Reeves and Clifford Nass. *The media equation : how people treat computers, television, and new media like real people and places*. Cambridge university press edition, 1996.

[20] David Sadek. *Attitudes mentales et interaction rationnelle : vers une théorie formelle de la communication*. Ph.D. thesis, Université de Rennes I, 1991.

[21] Aaron Sloman. Architectural requirements for human-like agents both natural and artificial. In Kerstin Dautenhahn, editor, *Human Cognition and Social Agent Technology (Advances in Consciousness Research)*. John Benjamins Publishing Co, 2000.

[22] Endel Tulving. *Elements of episodic memory*. Clarendon Press, Oxford, England, 1983.