Integrating Psychological Behaviors in the Rational Process of Conversational Assistant Agents

Jean-Paul Sansonnet
LIMSI-CNRS,
BP 133, 91403 Orsay Cedex, France
jps@limsi.fr

François Bouchet
SMART Laboratory, McGill University
3700 McTavish, Montreal, QC H3A 1Y2 Canada
francois.bouchet@mcgill.ca

Abstract

In this paper, we describe a framework dedicated to studies and experimentations upon the nature of the relationships between the rational reasoning process of an artificial agent and its psychological counterpart, namely its behavioral reasoning process. This study is focused on the domain of Conversational Assistant Agents, which are software tools providing various kinds of assistance to people of the general public interacting with computer-based applications or services. In this context, we show on some examples the need for the agents to be able to exhibit both a rational reasoning about the system functioning and a human-like believable dialogical interaction with the users.

Introduction

Adding psychological behaviors to rational agents

According to traditional definitions stemming from Artificial Intelligence and Multi-Agent Systems, Rational Agents are associated with programs capable of Practical Reasoning, *i.e.* building plans and choosing actions to be executed, in order to achieve their goals. For example, SOAR-based architectures are one of the first attempts at modeling the cognitive reasoning process of an agent (Laird, Newell, and Rosenbloom 1987) by means of explicit IF-THEN rules. More recently, the BDI approach of Bratman (1990), Rao and Georgeff (1995) is a theory of practical reasoning (deciding what to do next) directed towards situated reasoning about actions and plans (Allen et al. 1991). Recently, authors have proposed to integrate into rational agent architectures psychological notions, in order to propose: 1) a more complete cognitive models of agents; 2) agents capable of sustaining more human-like interactions with people, especially ordinary people involved in conversational activities with assistant agents. For example, Gratch and Marsella (2004) have proposed a model of emotions based on SOAR, with a significant impact upon the SOAR architecture. Using the agent creation platform JACKthat implements the BDI theory, CoJACK (Norling and Ritter 2004; Evertsz et al. 2008) is an extension layer intended to simulate physiological human constraints like the duration taken for cognition, working memory limitations (*e.g.* "loosing a belief" if the activation is low or "forgetting the next step" of a procedure), fuzzy retrieval of beliefs, limited focus of attention or the use of moderators to alter cognition. Emotions have also been integrated to the BDI framework, for instance with eBDI (Jiang, Vidal, and Huhns 2007) or KARO (Steunebrink, Dastani, and Meyer 2007). All those works provide a good introduction about the history of the necessity to implement emotions, and more generally psychological notions, into rational agents.

Although there has also been a lot of research works about the effect of personality on agents' behaviors in the virtual agents community (one of the most recent one being the SEMAINE project (Bevacqua et al. 2010)), they generally focus more on their impact on the animated agent (*e.g.* gaze or facial expressions) than on the rational decision process.

Adding psychological behaviors to agents

Psychological behaviors can play a major part in Conversational Assistant Agents (CAA), at the crossroad between:

Assistant Agents (Maes 1994), which are software tools designed to assist, in many ways, people involved in a computer-based or computer-mediated activity. The scope of application of assistant agents covers several roles such as: presenters, helpers, companions, teachers, coaches, *etc.* Research on assistant agents is based on artificial intelligence reasoning over symbolic representations and they focus on the notion of *rational agent*.

Conversational agents (Cassell et al. 2000), which are often embodied as virtual characters interacting with people through a dialogical session involving various modalities: textual or spoken natural language, body gestures, facial emotions, actions in the interface or environment, *etc.* Most conversational agents are given a personality and are hence supposed to interact with people according to the character they endorse: social role, personality traits, mental preferences, affects and moods. Research on conversational agents focuses on modeling *human psychology* (mental states, emotions, *etc.*) and its expression in conversational sessions.

Although presented above as separated notions, the rational and the psychological reasoning capacities of an agent actually work in quite an intricate manner (Ellsworth and Scherer 2003; Frijda 2006). Moreover, most studies mentioned in section focus on low-level/transitory psychological notions (such as emotions and moods, *e.g.* for natural language interaction (Allbeck and Kress-Gazit 2010)), while other notions associated with high-level/long-lasting

features of the personality of a human being (like Personality Traits (John, Robins, and Pervin 2008)) should also be integrated and be promising for developing CAA with *consistent* characters as defined by (Allbeck and Badler 2001).

A framework dedicated to experimental case studies on rational & psychological agents

We believe necessary to study the nature of relationships between the rational and psychological reasoning capacities of a CAA. It requires a dedicated test bed that we describe in this paper: the Rational and Behavioral (R&B) [1] architecture, where 'behavioral' here stands for 'psychological behavior'[2]. The R&B framework has been designed to meet two main requirements:

Genericity: in order to support various case studies, involving distinct viewpoints upon the rational/psychological interactions. Hence, languages and formats of the R&B architecture must act as a common layer upon which each case study can implement its particular strategies.

Separability: in order to prevent confusions between rational and behavioral concepts, and also to separate the competences of the designers, the design of the rational heuristics is separated from the design of the behavioral ones. Moreover, the execution of the two classes of heuristics is performed concurrently *on two separate engines*. Subsequently, the R&B framework makes it possible to experiment various kinds of agents personalities by combining, within a given agent, rational and behavioral heuristics, which were once elaborated separately.

Outline of the paper: The next section describes the general architecture of the R&B framework. Then we present the notations and languages working as a generic layer to support experimental case studies. In the final section, we present the implementation of agent's heuristics, illustrated by two distinct case studies showing the principles of genericity and separability of the R&B approach.

The R&B framework

Conversational Architecture

A typical R&B architecture, as illustrated in Figure 1, involves the following entities:
– the User `U`, who is an ordinary person who desires to use the system in the presence of a CAA,
– the System `S`, which can be for example a standalone application or an Internet service,
– the Agent `A`, which is a software tool endorsing a role in a given instance of given conversational situation.

The Graphical User Interface `GUI` is the traditional way for the user to interact with the system. However, when the user needs help, the Conversational User Interface `CUI` captures the multimodal interactions of the user with the agent and displays the reactions of the agent. The modalities handled by the the Dialogue manager `D` are classified into two

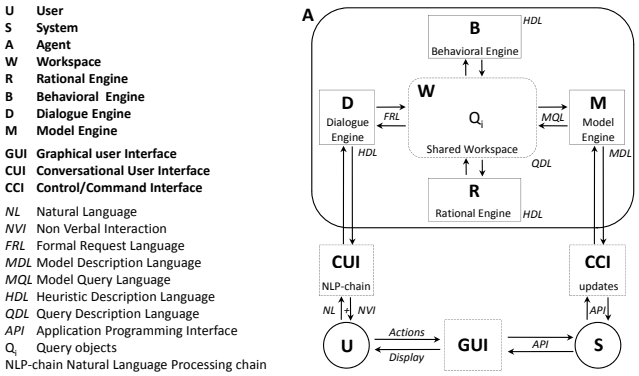

Figure 1: General architecture of an R&B agent.

main categories: 1) textual or oral Natural Language utterances (*NL*), that play a major part in assisting situations to ordinary people; and 2) other Non Verbal Interactions (*NVI*). Finally, the Control/Command Interface `CCI` links dynamically the symbolic model of the agent `M` to the system, so that the current state of the model and the current state of the system's runtime remain synchronized.

Internal structure of the agent

A typical agent, instantiated in the R&B framework, is made of five processing modules:

D: the Dialogue manager takes analyzed utterances from the `CUI` Natural Language Processing-chain (NLP-chain). It performs pragmatic reasoning heuristics (written in the so-called Heuristic Description Language, *HDL*) to produce formal requests expressed in the Formal Request Language (*FRL*).

M: the Model handles the symbolic representations of the agent $\mathcal{M} = \langle \mathcal{A}, \mathcal{U}, \mathcal{T}, \mathcal{S} \rangle$ where:
– $\mathcal{U}$ is the model of the user as a conversational character, containing data like its name, age, gender, *etc.*
– $\mathcal{A}$ is the mental model of the agent (*cf.* next section),
– $\mathcal{T}$ is the model of the the assisted application `S`,
– $\mathcal{S}$ is the model of the current dialogue session with `U`.
The symbolic structures in $\mathcal{M}$ are represented in the Model Description Language (*MDL*). they can be accessed in read/write mode by queries written in the Model Query Language (*MQL*). $\mathcal{M}$ is a dynamic structure evolving when updated by the agent reasoning engines (internal changes) or by the system via the `CCI` (external updates).

R: the rational reasoning engine of the agent implements the rational part of the role endorsed by the agent.

B: the behavioral reasoning engine of the agent implements the behavioral part of the role endorsed by the agent.

W: the engines `D`, `M`, `R` and `B` are considered as independent *processing engines* that work in parallel and communicate through a shared workspace `W` containing Query objects (Q_i) in Query Description Language (*QDL*).

[1] `http://www.limsi.fr/~jps/research/rnb/rnb.htm`

[2] The word 'behavioral' here is hence opposed to 'rational' - as such, we acknowledge it differs from the usual cognitive science terminology where rational behavior is just a subclass of behaviors

ask R|A|P	L asks for an information or checks if the proposition currently stands or if the action is known by the int.	Knowledge
tell P	L states an information or that the proposition currently stands	
reply V	L gives a value as an answer of request (ask,..) from the interlocutor	
know R|A|P	L states that he knows something about the content (when P, means he thinks it is true)	
unknown R|A|P	L states that he knows nothing about the content (or if P is true or false)	
mistrust P|V	L states that he thinks that the value is probably erroneous or the proposition is probably false	
why P|V	L asks why the proposition is currently standing or why the value has been replied	
possible A	L asks if it is possible (availability, rights...) for him or the agent to execute the action	
how A	L asks how to do the action/procedure	
effect A	L asks what will be the consequences of performing the action A	
execute A|R	L commands the interlocutor to execute the action or to activate the main function of the referenced object	Action
repeat -	L commands the interlocutor to execute again the last executed action	
undo -	L commands asks the interlocutor to undo the last executed action	
suggest A|P	L encourages/suggests/allows the interlocutor to execute the action / to adopt the proposition as a goal	
object A|P	L discourages/objects/forbids the interlocutor to execute the action / to adopt the proposition as a goal	
intent A|P	L states that he has the intention to execute the action in the near future / he has just adopted the proposition as a goal	
judge P	L expresses a subjective opinion by stating the proposition	Feelings
feel P	L expresses a subjective feeling by stating the proposition	
like R|A|P|V	L expresses a subjective preference/liking for the content (sub case of judge)	
dislike R|A|P|V	L expresses a subjective dis-preference/disliking for the content (sub case of judge)	
bravo -	L congratulates the interlocutor about the topic	
criticize -	L criticizes the interlocutor about the topic (e.g. contains abuse)	
agree -	L replies yes to a yes/no question or agree with the topic	Dialogue
disagree -	L replies no to a yes/no question or disagree with the topic	
greet	L greets the interlocutor at the beginning of the session	
bye	L asks for the session to end	

R reference
A action
P proposition
V value

'L' stands for the locutor (the user or the agent in replies)
'-' stands for the current focus of the dialogue session (IT)

Figure 2: List of main *FRL* performatives (left) with defining short-phrases (right)

Notations and languages

Formal Request Language (*FRL*)

In previous work (*cf.* footnote 2), we have implemented a NLP-chain dedicated to assistance-related interaction between novice users and CAAs. This work made it possible to collect a domain-oriented corpus of 11 000 help utterances and it enabled the analysis of the linguistic domain related to the function of assistance (Bouchet and Sansonnet 2009b). In this paper, as we focus on the agent's architecture, we will rely on a simplified version of the Formal Request Language (*FRL*), as shown in Fig. 1; we just give the main notations, leaving aside complex cases like reported speech, conditional commands, past/future, not to speak of grounding issues, input noise *etc.*

A basic *FRL* request is of the form $F_{loc}(X)$ where:
– F is the performative (not unlike the DAMSL approach to speech acts (Core and Allen 1997)) or, of one of the four classes: **K**nowledge, **A**ction, **F**eeling, **D**ialogue. When necessary, loc indicates the locutor (the user `U` or the agent `A`).
– X is the content, of one of the four classes: **R**eference, **A**ction, **P**roposition, **V**alue.

The model of the agent

Structure Basically, the model $\mathcal{M}$ is an evolving tree structure (as in evolving algebra for abstract state machines (Gurevich 1995)). Given a new session, the model $\mathcal{M}_0$ starts at t_0: $\mathcal{M}_0 = \langle \mathcal{A}_0, \varnothing, \mathcal{T}_0, \varnothing \rangle$, where $\mathcal{A}_0$ is the submodel of a given agent and $\mathcal{T}_0$ is the submodel of a given application. $\mathcal{M}$ non terminal nodes are labelled by concepts (a concept is a symbol or an index) and terminal nodes are conventional values (Symbols, Numbers, Booleans, Strings).

Table 1: Main *MQL* functions

Query name	Query action
`GET[path]`	returns the subtree from `n`
`COUNT[path, expr]`	returns the number of subtrees from `n`with value `expr`
`SET[path, expr]`	replaces `n` by `expr`
`MAP[path, func]`	replaces `n` by func(`n`)
`ADD[path, expr]`	appends `expr` to `n`
`DEL[path,` s_i`]`	deletes subtree of head s_i in `n`
`VOID[]`	does nothing

Model Query Language (*MQL*) The model is accessed by the agent or by the application using the Model Query Language (*MQL*). The main access functions, used in the examples of the last section, are described in Table 1, where `path` stands for a tree path expression $\mathcal{M}.s_1.s_2,..,s_n$ (s_i being node labelling symbols), `n` is the node referred to by `path` and `expr` is a terminal value or a subtree. The replies are of the form `OK[result]`, or `FAIL[report]` if it fails.

Mental model of the agent

Of the four submodels of $\mathcal{M}$, the most specific to this paper is $\mathcal{A}$ that supports the representations of the agent's mind. The R&B framework can support various mental models, defining various agents, provided they are expressed in the formalism of the model $\mathcal{M}$. As an example, we define here a specific mind model supporting the case studies presented in the last section. It covers most significant notions discussed in the mental states modeling literature (Ortony 2003) de-

Table 2: The four types of agent's mental states

	Arity	
Dynamicity	Unary	Binary
Static	Trait Ψ_T	Role Ψ_R
Dynamic	Mood Ψ_m	Affect Ψ_a

spite some simplifications (*e.g.* we consider traits and roles are static during a dialogue session). This model distinguishes four types of mental states according to their dynamicity and to their arity, as summarized in Table 2. Each of them is associated with a value in $[-1.0, 1.0]$, where 1.0 denotes the maximum intensity of the concept, -1.0 is the maximum intensity of the antonym of the concept and 0 stands for the "neutral" position (neither the concept nor its antonym stand).

Unary categories: the agent is viewed as autonomous:

Traits (Ψ_T) correspond to typical personality attributes that can be considered as stable during the agent's lifetime, implemented using the classical "Five Factors Model" of personality traits (Goldberg 1981):
– *Openness*: appreciation for adventure and curiosity,
– *Conscientiousness*: self-discipline, will to achieve goals,
– *Extraversion*: energy, strength of positive emotions and tendency to seek company of others,
– *Agreeableness*: being compassionate and cooperative,
– *Neuroticism*: tendency to feel negative emotions.

Moods (Ψ_m) are agent's factors varying with time thanks to heuristics and according to previous state of the agent and to the current state of the world. We define:
– *Happiness*: physical contentment wrt current situation,
– *Satisfaction*: cognitive contentment wrt current situation,
– *Energy*: agent's physical strength,
– *Confidence*: agent's cognitive strength.

Binary categories: (also called interpersonal categories) the agent `A` interacts with another actor, called `U`.

Roles (Ψ_R) represent a static relationship between the agent and `U`. We define two main categories of roles:
– *Authority*: the right the agent feels to be directive and reciprocally not to accept directives from others This role is often antisymmetric, *i.e.*: `Authority(X,Y) = -Authority(Y,X)`
– *Familiarity*: the right the agent feels to use informal behaviors towards `U`. This role is often symmetric.

Affects (Ψ_a) denote, in this particular model, dynamic relationships between the agent and `U`. We define three kinds of affects:
– *Dominance*: the agent feels powerful relatively to `U`. This relationship is often antisymmetric.
– *Cooperation*: the agent tends to be nice, and helpful with `U`. It is not necessarily symmetric.
– *Trust*: the agent feels it can rely on `U`. It is not necessarily symmetric.

Shortened notation: The actual value of a mind attribute like "happiness" can be accessed by its full path in the model tree ('$\mathcal{M}.\mathcal{A}$.mind.mood.happiness.val') or by using a shortened notation of the mind model (M_h). Moreover, although in the model attributes values $v \in [-1, 1]$, it is often more convenient to consider a five-level scale based on a discrete partition of the domain $[-1, 1]$ into five contiguous intervals: $< - = + >$ (*e.g.* $>$ is $[0.8, 1]$), and intervals can be grouped by juxtaposing them, *e.g.* $M_h^{+>} \wedge A_c^{<}$ means that the agent is happy or strongly happy and completely antagonistic.

Query Description Language (*QDL*)

A query is an element of `W` that *wraps* a request written in *FRL* or *MQL* and provides extra attributes. It has the following structure and shortened notation:

$$Q_i = [\text{val}[\{r|\{r_i\}], \text{history}[\{\texttt{D}, \texttt{R}, \ldots\}], \text{to}[\texttt{M}], \text{status}[+]]$$
$$= Q_i.\text{val}^{Q_i.\text{status}}_{Q_i.\text{history}|Q_i.\text{to}} = \{r_1, \ldots, r_n\}^{+}_{\{\texttt{D},\texttt{R},\ldots\}|\texttt{M}}$$

Where:
– i $\in \mathbb{N}^+$ absolute identifier of a query
– val contains one or a sequence of *FRL* |*MQL* requests
– history $\in \{\texttt{D}, \texttt{R}, \texttt{B}, \texttt{M}\}^*$ stack of engines that handled Q_i
– to $\in \{\texttt{D}, \texttt{R}, \texttt{B}, \texttt{M}\}$ next engine meant to treat Q_i
– status $\in [\emptyset, -, +]$ success status of Q_i

Note that although a query Q_i can be given a destination (field 'to'), it doesn't prevent other engines to access Q_i while it is in the workspace `W` and to possibly alter it.

Implementation of the heuristics

Heuristic Description Language (*HDL*)

HDL makes it possible for both rational and psychological designers to handcraft rules. The main reason for this choice is that the R&B framework is dedicated to experimental studies: designers will have to share and read heuristics from others (*e.g.* see examples proposed below); besides we are not at the stage where a rule-learning process (inductive or other) can be easily implemented.

Syntax A heuristic defines a rational or behavioral reaction to a class of formal requests expressed in *QDL* (defined by a pattern matching expression). Its general form is:
H : id[QDL pattern]:–{GuardedScript$_1$, …, GuardedScript$_n$}
Where:
GuardedScript $\equiv$ {Guard$_1$ $\rightarrow$ Script$_1$, …, Guard$_n$ $\rightarrow$ Script$_n$}
Guard$_i$ $\equiv$ Logical expr | $\emptyset$ ($\emptyset$ = True)
Script$_i$ $\equiv$ Instruction | {Instruction$_1$, …, Instruction$_n$}
Instruction$_i$ $\equiv$ Basic operation | Query call | GuardedScript
Query call $\equiv$ Q[Query id, {*FRL* req — *MQL* req}]
Note that instructions can recursively be guarded scripts.

Dynamics In the R&B framework, for a given case study, a set of heuristics can be defined and associated with any of the four engines `D`, `M`, `R` and `B` (here, we only discuss those associated with `R` and `B`). Their execution is performed by the Heuristic Scheduler (HS) which ensures their coroutining and achieves its control at two levels:
– within a given heuristic H, it decides when instructions (guard→script) should be executed,
– within a given R&B case study, it decides when engines and heuristics should take a turn.

As guards in heuristics can overlap, several *execution policies* can be selected (*e.g.* first-hit-exit, execute-all, random-choice…), and as a guard can remain active ($true$)

after the execution of its script, several *repetition policies* can also be selected (*e.g.* execute-once, loop-over). Moreover, since in the same engine or in distinct ones, several heuristics can match one query or several queries situated into W, again, several *heuristic policies* (behavior-first, rational-first, alternate-M/B) and *query policies* (FIFO-based, random choice...) are possible.

The principle of genericity, stated in the first section, compels the R&B scheduler to be parametrizable to enable various simulations. For the case studies presented below, we will use a single scheduling policy, such that:

– *Within heuristics:* all instructions with active guards ($true$) are executed; when several guards are simultaneously active, a random choice is performed; instructions are only executed once; a heuristic is terminated when all its instructions are executed (which may never happen – hence, W is cleared after each request handling).

– *Between heuristics:* all heuristics that match a query object in W are launched (*i.e.* coroutined with the already launched ones). When a heuristic is terminated, it can be launched again (but no reentrance is available). When several heuristics (even associated with different engines) are eligible, a random choice is performed, thus resulting in various R&B interleaved executions (*cf.* last example).

Case study 1: Asking for information

Assume the user puts the question "What is your age?", resulting in the addition to the workspace of $Q_1 = \{\text{ASK}_u[\text{agent.age}]\}^{\emptyset}_{\{\text{D}\}|R}$

A simple rational reaction: A possible rational heuristic that can handle questions about the agent's attributes is:

1: H_{R1} : ask-agent-attribute[$\{\text{ASK}_u[\text{agent.x\_}]\}^{\text{-}}_{\text{-}|\text{-}}$] :–{
2: $\rightarrow Q[i, \texttt{GET[x\_]}]$,
3: $Q_i^+ \rightarrow Q[j, \text{TELL}_a[\text{agent.x\_}, Q_i.\text{val}]]$,
4: $Q_i^- \rightarrow Q[j, \{\text{UNKNOWN}_a[\text{agent.x\_}],$
: $\text{TELL}_a[Q_i.\text{val}]\}]$
5: $Q_i^{\neg\emptyset} \rightarrow Q^+_{\text{this}}$
6: }

Explanations:
1: x_ is a pattern variable matching any symbol like age, gender...
2: The empty guard prompts the script to be executed immediately. In Q_1, x_ being 'age' (shortcut for full path $\mathcal{M}.\mathcal{A}$.age.val), it creates a new query Q_i to retrieve this value from the model.
3: If the request in Q_i has been successful (Q_i.status == $+$), *FRL* request TELL_a[agent.x_, Q_i.val] is wrapped into a new query Q_j, and Q_i.val contains a *MQL* request OK[retrieved-val].
4: If the request in Q_i has been unsuccessful (Q_i.status == $-$), a *FRL* answer in two parts is wrapped into a new query Q_j and Q_i.val contains FAIL[report].
5: Once Q_i has been handled, the current query Q_{this} is declared to have been successfully handled as well.

In any case, the request in Q_j is then retrieved by D to be sent to U.

Handling user's repetitions: If the same request (in *FRL*) is issued several times during the same dialogical session, since a new wrapping query will be created for each of them, H_{R1} will generate exactly the same formal answer. But lack of handling of repetions has been identified as a major cause of the lack of human-likeness (Xuetao, Bouchet, and Sansonnet 2009) in CAA: rationality isn't enough. A solution consists in using a simple rational heuristics to actually store the interaction with the user, like:

H_{R2} : interact-mem[$\{(\text{x\_})_u[\text{y\_}]\}^{\text{-}}_{\text{-}|\text{-}}$] :–{
$\rightarrow Q[i, \texttt{ADD[}\mathcal{S}, \text{x\_[y\_]]}]\}$

And a behavioral one generating additional *FRL* and *MQL* queries, such as:

1: H_{B1} : repetition[$\{(\text{x\_})_u[\text{y\_}]\}^{\text{-}}_{\text{-}|\text{-}}$] :–{
2: $\rightarrow Q[i, \texttt{COUNT[}\mathcal{S}, \text{x\_[\_y]}]$
3: $Q_i^+ \wedge Q_i.\text{val} > 1 \rightarrow Q[j, \text{TELL}_a[\text{repetition}]]$
4: $Q_i^+ \wedge Q_i.\text{val} > 2 \rightarrow Q[k, \texttt{MAP[}\text{coop}, \lambda\, x.x * 0.9\texttt{]}]$
5: $Q_i^+ \wedge Q_i.\text{val} > 4 \rightarrow Q[j, \text{DISLIKE}_a[\text{repetition}]]$
6: }

Explanations:
2: Retrieval of the number of similar requests previous issued.
3&5: Extra information to the user reveals increasing boredom.
4: The agent modify its mind state in an appraisal-like reaction (coop $\equiv \mathcal{M}.\mathcal{A}$.mind.mood.cooperation.val), with $\lambda\, x.x * 0.9$ being a λ-expression decrementing its argument by 10%.

Case study 2: Handling user's feelings

Assume that the user hasn't been satisfied by the agent previous reaction(s) and now expresses only her/his feelings about it with a force that can range on a scale from "I am not satisfied" to "I hate you!". It results in the same class of *FRL* request, generating the query: $Q_2 = \{\text{DISLIKE}_u[\text{agent}\}^{\emptyset}_{\{\text{D}\}|R}$

A simple behavioral reaction: Dealing with an emotional reaction can't be rational and "objective", and a possible behavioral reactions could be given by a heuristic like:

1: H_{B2} : dislike-agent[$\{\text{DISLIKE}_u[\text{agent}]\}^{\text{-}}_{\text{-}|\text{-}}$] :–{
2: $\rightarrow$ { $Q[i, \texttt{MAP[}\text{energy}, \lambda\, x.x * 0.9\texttt{]}]$,
3: $Q[j, \texttt{MAP[}\text{confidence}, \lambda\, x.x * 0.9\texttt{]}]$,
4: $Q[k, \texttt{MAP[}\text{cooperation}, \lambda\, x.x * 0.9\texttt{]}]$ }
5: $Q_i^+ \wedge Q_i.\text{val} < -0.5$
: $\rightarrow Q[l, \text{TELL}_a[\text{energy}, \text{“tired”}]]$
6: $Q_j^+ \wedge Q_j.\text{val} < -0.5$
: $\rightarrow Q[l, \text{TELL}_a[\text{confidence}, \text{“depressed”}]]$
7: }

Explanations:
2–4: Executes a sequence of queries to modify agent's mind state.
6: If its energy is very low, a query with a *FRL* request to say "I feel tired" is generated.
7: A second *FRL* request can be added to that query (if it already exists, to a new one if not).

Taking mood into account: Despite the purely emotional aspect of Q_2, some rationality is still necessary, at least to remember user's negative opinion of the agent with a heuristic like H_{R2}, we have:

H_{R3} : dislike-mem[$\{\text{DISLIKE}_u[\text{x\_}]\}^{\text{-}}_{\text{-}|\text{-}}$] :–{
$\rightarrow Q[i, \texttt{ADD[}\mathcal{U}.\text{dislikes}, \text{x\_]}]\}$

However, a basic rational reaction like that can be put into question by the agent's current mind state. For instance, if the agent is currently in a high level of satisfaction and not neurotic $M_s^{>} \wedge T_n^{<-=}$, it would tend to be in denial when

facing negativity into user's utterances. A behavioral alteration upon this query put in W then could be:

1: H_{B3} : good-mood[{ADD[($\mathcal{A}$ —x).dislikes, x_]}$^{-}_{-|-}$] :–{
2: $M_s^{>} \wedge T_n^{<-=} \rightarrow Q^{+}_{\text{this}} \wedge Q_{\text{this}}$.val=VOID[]
3: ...}

Explanations:

1: The agent refuses a *MQL* query adding a dislike into $\mathcal{M}$.

2: So it replaces it by a *MQL* request VOID[] and declares the query as successfully handled.

Note that, depending on the order in which the heuristics are applied, several sequences can be produced: $<H_{R3}, H_{B3}, H_{B2}>$ or $<H_{B2}, H_{R3}, H_{B3}>$, thus resulting in a variety of reactions that in turn can be perceived by the users from simple variants to drastically different behaviors.

Implementation and conclusion

In previous works, a full CAA architecture has been implemented[3], which encompasses the components of the R&B architecture defined in Figure 1. Moreover, it has enabled us to collect a corpus of assistance-based natural language utterances that resulted in the grounding of the *FRL* language (Bouchet 2009). Recently, a first toolkit to experiment R&B agents has been implemented (Bouchet and Sansonnet 2009a) in Mathematica. This toolkit can be freely accessed at the Web page of the R&B project (*cf.* footnote 1).

We have proposed a framework dedicated to the support of experimental case studies about the R&B problem: the nature of the relationships between rational and psychological reasoning. The particular context of the conversational assistant agents was chosen because this issue is central to the acceptability factor of those systems, hence providing a good test field. In future work, more case studies must be performed to confirm that the R&B framework actually provides a generic layer to experiment various strategies for integrating psychological behaviors into rational agents, as well as a validation using human subjects to evaluate the identification of implemented behaviors.

References

Allbeck, J. M., and Badler, N. I. 2001. Towards behavioral consistency in animated agents. In *Proceedings of the IFIP TC5/WG5.10 DEFORM'2000 Workshop and AVATARS'2000 Workshop on Deformable Avatars*, 191–205. Kluwer, B.V.

Allbeck, J. M., and Kress-Gazit, H. 2010. Constraints-based complex behavior in rich environments. In *Intelligent Virtual Agents (IVA 2010)*, volume 6356 of *LNAI*, 1–14. Philadelphia, PA: Springer-Verlag.

Allen, J. F.; Kautz, H. A.; Pelavin, R. N.; and Tenenberg, J. D. 1991. *Reasoning About Plans*. Representation And Reasoning. Morgan Kaufmann Publishers, Inc.

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

Bouchet, F., and Sansonnet, J. P. 2009a. A framework for modeling the relationships between the rational and behavioral reactions of assisting conversational agents. In *Proc. of the 7th European Workshop on Multi-Agent Systems (EUMAS'09)*.

Bouchet, F., and Sansonnet, J. P. 2009b. Tree-kernel and feature vector methods for formal semantic requests classification. In Perner, P., ed., *Machine Learning and Data Mining in Pattern Recognition*, 126–140. Leipzig, Germany: IBaI Publishing.

Bouchet, F. 2009. Characterization of conversational activities in a corpus of assistance requests. In Icard, T., ed., *Proc. of the 14th Student Session of the European Summer School for Logic, Language, and Information (ESSLLI)*, 40–50.

Bratman, M. E. 1990. What is intention? In Cohen, P. R.; Morgan, J.; and Pollack, M. E., eds., *Intentions in Communication*. The MIT Press. 15–32.

Cassell, J.; Sullivan, J.; Prevost, S.; and Churchill, E., eds. 2000. *Embodied Conversational Agents*. MIT Press.

Core, M. G., and Allen, J. F. 1997. Coding dialogs with the DAMSL annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, 28–35.

Ellsworth, P. C., and Scherer, K. R. 2003. Appraisal processes in emotion. In Davidson, R. J.; Scherer, K. R.; and Goldsmith, H. H., eds., *Handbook of affective sciences*, Series in affective science. New York, NY, USA: Oxford University Press. 572–595.

Evertsz, R.; Ritter, F. E.; Busetta, P.; and Pedrotti, M. 2008. Realistic behaviour variation in a BDI-based cognitive architecture. In *Proc. of SimTecT'08*.

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

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

Gratch, J., and Marsella, S. 2004. A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research* 5(4):269–306.

Gurevich, Y. 1995. Evolving algebras 1993: Lipari guide. In *Specification and validation methods*. Oxford University Press, Inc. 9–36.

Jiang, H.; Vidal, J. M.; and Huhns, M. N. 2007. eBDI: an architecture for emotional agents. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 1–3. New York, NY, USA: ACM.

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

Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. Soar: an architecture for general intelligence. *Artif. Intell.* 33(1):1–64.

Maes, P. 1994. Agents that reduce work and information overload. *Commun. ACM* 37(7):30–40.

Norling, E., and Ritter, F. E. 2004. 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*.

Ortony, A. 2003. On making believable emotional agents believable. In Trappl, R.; Petta, P.; and Payr, S., eds., *Emotions in humans and artifacts*. Cambridge, MA: MIT Press. 189–211.

Rao, A. S., and Georgeff, M. P. 1995. BDI agents: From theory to practice. In *Proc. First Int. Conference on Multi-agent Systems (ICMAS-95)*, 312–319.

Steunebrink, B.; Dastani, M.; and Meyer, J. 2007. A logic of emotions for intelligent agents. In *Proc. of the National Conf. on Artificial Intelligence*, volume 22, 142.

Xuetao, M.; Bouchet, F.; and Sansonnet, J. P. 2009. Impact of agent's answers variability on its believability and human-likeness and consequent chatbot improvements. In *Proc. of AISB 2009*, 31–36.

[3] http://www.limsi.fr/~jps/research/daft