2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

Influence of personality traits on the rational process of cognitive agents

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

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

***Abstract*—In this paper we present an approach based on the principle that psychological capacities, especially personality traits, influence the decision making process of rational agents. While using the FFM/NEO PI-R taxonomy, we propose a model for the expression of personality traits in terms of so-called influence operators that add meta control rules to the cycle of rational BDI agents.**

***Keywords*- Agent cognitive modeling, Computational personality, Personality traits, Rational and Psychological agents.**

I. Introduction

A. Rational and psychological agents

Rational Agents: Many agent cognitive architectures are based on practical reasoning, in the following of the SOAR and ACT-R frameworks, or more recently BDI-agents [25]. In Bratman's theory of practical reasoning, an agent's behavior is modeled by specifying *beliefs*, *goals* and *plans* and is effectively produced through the agent's deliberation cycle. These architectures have been quite successful at creating both autonomous and multi-agent systems capable of operating in computational contexts.

Conversational Agents: However, agents are more and more in interaction with human users, as revealed by the growing of so-called conversational agents [5]. In this new context, authors have claimed agents should be both *competent* (thanks to their symbolic reasoning capacities [28]) and psychologically *relevant*, in order to increase: a) their acceptability factor, especially when they deal with people of the general public, b) the efficiency factor of the agent/human interactional process itself (*e.g.* for teaching).

Psychological Agents: Actually, since the introduction of the notion of 'Believable Agents' in the mid-90's [2], [19], [15] there has been various attempts at implementing psychological features into cognitive architectures. For example, the works of Rousseau and Hayes-Roth [27] have established a first ground by providing examples of how personality factors (also called personality traits) can be implemented into the cognitive architecture of a rational artificial agent. All these authors emphasize the fact that agents should exhibit both a rational reasoning engine and a psychological reasoning engine.

B. Categorization of personality traits

Personality traits taxonomies When one is interested in the taxonomy of the psychological phenomena, especially those related to personality traits, the most successful paradigm to day is the Five Factor Model (FFM - also named OCEAN), which is a convergent research from many authors in Psychology during the last 20 years, based on five large classes of traits. FFM has taken upon Cattell's classification into 16 factors [6], which was still prominent in the 80s and was supported by Eysenck's Personality Questionnaires (EPQ) [11]. FFM is mainly based on natural language lexical resources such as the glosses found in dictionaries [13], according to the 'lexical hypothesis', which states that most of socially relevant and salient personality characteristics have become encoded in the natural language [1].

Traits, Facets and Schemes However, classifications such as Cattell's 16 factors or FFM are very generic, which led some authors to propose to refine these taxonomies with so-called *facets*. For example, the NEO PI-R facets [7], have been proposed for FFM, thus resulting in a more precise two-level taxonomy (traits/facets) of 30 bipolar positions. It remains nonetheless too generic from a strict computational viewpoint, as implementing psychological behaviors in software agents requires operational definitions. This led us, in recent works [3], to develop a resource[1] based on an enrichment of the FFM/NEO PI-R taxonomy with glosses associated with the senses of a large set of 1055 personality adjectives, while using the WordNet lexical data base [12]. The glosses have been completed and aligned with 300 Goldberg's questionnaire so-called *q-items*[2]. Finally for each FFM/NEO PI-R position, glosses and items have been clusterized in so-called *behavioral schemes* (in short schemes) associated with congruent operational behaviors (An excerpt is given in Figure 1 for trait Conscientiousness).

C. Computational expression of personality traits

Our objective is to provide rational agents with psychological capacities. Our approach is based on the computational expression of personality traits, in terms of their *influence*

[1]This resource is available on the Web and freely downloadable at: http://www.limsi.fr/~jps/research/rnb/toolkit/taxo-glosses/taxo.htm
[2]http://ipip.ori.org/newNEOKey.htm

978-0-7695-4513-4/11 $26.00 © 2011 IEEE
DOI 10.1109/WI-IAT.2011.170

over the decision making process of the agents. To this purpose, we have developed a specific framework composed of three main parts described in the next sections:

— *Rational agent model* It is based on the typical deliberation cycle of BDI agents, which is decomposed in five steps for the purpose of this study. Section II-A presents the five steps and Section II-B describes a basic plan language, underlying the rational process.

— *Influence operators* Given a typical rational agent model, our first issue is to substantiate the claim that it can be influenced by meta heuristics implementing personality traits. This entails that the deliberation cycle should exhibit capabilities for being influenced, according to various modalities. A first result of this study is that it has been possible for us to elicit eight distinct classes of so-called *influence operators*, which add meta control over the deliberation cycle upon actions and plans. They are listed and described in Section III.

— *Trait implementation* Given the rational model and the eight classes of influence operators, we have a) to define a psychological agent model and b) to show how the psychological model can be implemented through influence operators. In Section IV-A, we sketch a generic psychological agent model then we restrict the discussion to its static part *i.e.* the computational expression of the personality traits. In this study, we have chosen the FFM/NEO PI-R taxonomy enriched with the schemes defined in Section I-B. Actually, schemes provided operational behaviors that appeared to be quite easily transposable in terms of so-called *activation levels* for the influence operators. Section IV presents a case study about the implementation of the schemes present in the particular FFM trait Conscientiousness.

D. Related work

Gratch and Marsella [14] have implemented a model of emotions based on traditional SOAR architecture but most authors have proposed improvements of BDI architectures exhibiting both rational reasoning modules and psychological reasoning modules [18]. For example, the eBDI model of Jiang et al. [17] implements emotions in a BDI framework. They give a good introduction about the necessity to implement emotions into rational agents.

Gratch and Marsella works showed that SOAR architecture does not easily support the implementation of psychological behaviors and that the SOAR core engine must be adapted. Alternatively, BDI architectures offer a more open and flexible engine (the deliberation cycle) hence we rely on it for the support of our framework. For example, one can rely on frameworks like 2APL, which is a BDI-based agent oriented programming language developed by Dastani [8] providing practical programming constructs to allow the generation, repair, and (different modes of) execution of plans based on beliefs, goals, and events.

However our approach is distinct from most above mentioned studies using BDI engines, mainly because in those studies the psychology of the agent is based on dynamic mental states (like Moods Φ_m and Affects Φ_a, defined in Section IV-A), which influence the expression of emotions –often only through the body modalities of a virtual character– but they have no or little impact upon the decision making process of the agent. Instead we propose here an approach where the *static* features of the personality of an agent are expressed through its action (Note: in our approach, expression through body modalities is actually handled by modal operators defined in Section III-B).

Using the well-used agent creation platform JACK [16] that implements the BDI theory, CoJACK [10] 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. A similar approach is taken for conversational agents in the PMFserv framework [31]. In these studies, authors focus on the influence of physical or cognitive capacities over the deliberation cycle but not on actual psychological, like moods or traits.

Closer to our work, Malatesta et al. [20] use traits to create different expressions of behaviors, especially by influencing the appraisal part of the OCC theory [23]. The difference is that their work focuses on how agents evaluate the results of their actions and of external events, not on the way they perform a task. Nonetheless the idea that traits can differentiate agents' behaviors underlies this work.

In the same way, Rizzo et al. [26] have shown that goals and plans can be used to represent a character's personality in an efficient way, by attributing specific behaviors to the pursuit of each goal. Personality traits are used to choose between the multiple goals of a BDI agent (*i.e.* traits influence Desires). Once chosen, goals are planned and executed directly whereas in our case, traits operate on already planned goals (*i.e.* traits influence Intentions). This applies also to works of Pelachaud et al. [21], based on the well-used architecture of conversational agent GRETA [24], which involves models of personality for the expression of emotions (face, gesture, *etc.*). It applies also to the FATIMA architecture [9] stemming from [24], which implements personality traits.

II. Agent Model

A. Agent cycle

Using an approach similar to a BDI-agent cycle process [25][30], we will divide the agent rational reasoning process in a five-step cycle:

1. Deliberation As in BDI theory, given an agent $\mathscr{A}$ with an initial set of possibly contradictory 'desires' $\gamma_i \in \mathbb{G}$, during the deliberation step, the agent chooses within its desires a subset of so-called goals γ_i^* that: 1) it intends to achieve, and 2) must be non contradictory.

2. Planning Given the current goal γ^* in γ_i^*, the agent generates a rational plan $\pi^* \in \mathbb{P}$, which has γ^* in its postconditions: $\gamma \in post(\pi^*)$. A plan is a symbolic expression of the plan language $\mathscr{L}_\pi$ defined in Section II-B. It is composed of subplans $\pi_i \in \mathbb{P}$ and terminal actions $\alpha_i \in \mathbb{A}$.

3. Decision Although the planner proposes a single plan expression π^* in order to achieve the current goal γ^*, this expression often contains alternatives (equivalent subplans or equivalent terminal actions) and options (optional subplans or optional terminal actions) defined in Section II-B. Hence, plans are subdetermined and the agent has to make choices between alternatives and to decide whether to execute or not options. Other kinds of decisions are required in the scheduling of the plan operators (discussed in Section II-B). In any case, the result of the decision process is to elicit the next action α^* of π^* to be executed.

4. Execution The agent executes the current action α^*. In most BDI architectures, this step is implicit: it is explicitly given here in order to support the influence of psychology traits over the execution process itself.

5. Evaluation The agent receives, from the application, information about the actual results of the execution of α^*. It is then possible for the agent to evaluate its performance, both in rational and psychological terms.

B. Plan language

The plan language $\mathscr{L}_\pi$ defines expressions that are composed of two kinds of entities: sub plans and terminal actions. In turn, terminal actions are defined in terms of application dependent functions. In the following, when there's no ambiguity, we'll simply refer to sub plans as plans, to terminal actions as actions, and to application dependent functions as functions.

1. plans are expressions of the form:

$$\pi = \langle p_i, \gamma_i, pre, post, attr, body \rangle$$

where:

– p_i is a unique plan identifier. Below, we refer to a plan either as π or p_i.

– *pre* is the set of rational preconditions of the plan (noted $pre(p_i)$), *i.e.* facts that *must* hold in the world for π to be applicable.

– *post* is the set of rational postconditions of the plan (noted $post(p_i)$), *i.e.* facts that *necessarily* or *possibly* hold in the world after the execution of π (*cf.* Section II-B-4).

– $\gamma_i \subseteq post(p_i)$ is the set of goals *of the plan*. It is important to note that the agent may not have the intention to make all the postconditions hold after plan execution, but only the γ_i ones (*cf.* the strategic bomber scenario of Bratman [4]).

– *attr* is the set of attributes defining the plan *wrt* psychological influences.

– *body* implements the plan in terms of sub plans and terminal actions (*cf.* Section II-B-5).

Table I
PROCEDURAL OPERATORS BY DESCENDING ORDER OF PRECEDENCE.

Name	Op	Semantics (informal)
seq	;	$a_1;a_2$: Done(a_1) is a precondition to start a_2
alt	|	$a_1 \| a_2$: only one element is randomly executed
par	||	$(a_1 \| a_2) \equiv (a_1;a_2)\|(a_2;a_1)$ one element of the sequence is randomly chosen and executed
case	$\mapsto$	guard$_1 \mapsto a_1$, guard$_2 \mapsto a_2$ guard$_i$ are explicit preconditions for a_i to be executed If several guards are True, one guard is randomly chosen and executed

2. actions are expressions of the form:

$$\alpha = \langle a_i, \gamma_i, pre, post, attr, call \rangle$$

where:

– a_i is a unique action identifier.

– *call* is a call to a function $f_i \in \mathbb{F}$, executable in the world. Other components are defined as in plans above.

3. functions are application dependent basic operations. They are defined in a part of the model dedicated to the application, through an API (namely $\mathbb{F}$) of the form:

$$\varphi = \langle f_i, args, params \rangle$$

where:

– f_i is a unique function identifier, provided by the API of the application.

– *args* is the set of the description of the rational arguments of function f_i. Rational arguments correspond to the so-called 'functional proprieties' of the function.

– *params* is the set of the description of the modal arguments of the function. Modal arguments correspond to the so-called 'non functional proprieties' of the function (*e.g.* speed, focus of attention *etc.*). We state that $args \cap params = \emptyset$.

4. postconditions are facts that hold in the world after the execution of an action or a plan. They can be divided in two main classes:

– Necessary postconditions (noted $post_N$) are sure to hold. For example, given an action a_0 linked to function throw-a-dice, $upface = i \;\; i \in [1..6]$ is a member of set $post_N(a_0) \subseteq post(a_0)$;

– Possible postconditions ($post_P$) are not sure to hold but they could. For example, given a_0, $upface = 6$ is a member of set $post_P(a_0) \subseteq post(a_0)$.

5. plan body the body of a plan has the following syntax:

Body	$\rightarrow$ BodyProc | BodyDecl | a_i
BodyProc	$\rightarrow$ ProcOp[Body$\|p_i\|a_i, \ldots$]
BodyDecl	$\rightarrow \langle S_p, S_o, S_d \rangle$
$S_p \| S_o \| S_d$	$\rightarrow \{p_i \| a_i, \ldots\}$

where:

– BodyProc is composed with procedural operators (ProcOp) as defined in Table I (given in infix notation).

– BodyDecl is a declarative expression composed of three subsets of subplans or actions: S_p is the set of preferred elements; S_o is the set of optionally executable elements; S_d is the set of default elements.

Table II
INFLUENCE OPERATORS' OPERANDS AND STEP OF APPLICATION.

Class	Operand	Step of application
Goal	γ	Deliberation (1)
Preference	$\alpha\|\pi$	Decision (3)
Filter	$\alpha\|\pi\|\gamma$	Deliberation or Decision (1 or 3)
Scheduling	π	Execution (4)
Modal	$\varphi\|\alpha\|\pi$	Execution (4)
Option	α	Execution (4)
E**X**pectation	$\alpha\|\pi\|\gamma$	Evaluation (5)
Appraisal	$\alpha\|\pi\|\gamma$	Evaluation (5)

Plans are synthesized by a rational planner, thus entailing two consequences upon the semantics of BodyDecl expressions. 1) Since S_p, S_o and S_d are sets, their elements are considered by the rational planner as equivalent alternatives (within a set), at least *wrt* to the goals associated with the body, if not *wrt* to all postconditions. 2) The meaning attached to 'preferred', 'optional' and 'default' is restricted to the domain of rational planning and is not related to psychology (*e.g.* for 'preferred' see footnote[3]).

III. DEFINITION OF INFLUENCE OPERATORS

A. Definition

An influence operator $\omega \in \mathbb{I}$ is of the form:

$$\omega = \langle o_i, rules, level \rangle$$

Where:

– o_i is a unique operator identifier of the form C_{name}, C being a letter associated with the *class* of the operator (*cf.* below).
– *rules* is a set of (meta) rules adding control over the five steps of the rational process cycle of the agent (*cf.* Section II-A). They define the operational semantics of the operators as operations on: goals, actions and plans.
– *level* is the level of activation of the operator, which defines how the operator must be applied over the cycle in order to implement a particular trait of personality (see Table IV). A level is composed of:

A force $v \in \{0,1,2\}$, used to prioritize the operator (0 means not activated, 1 is activated with low priority (when several operators apply) and 2 is activated with high priority). If different from 0, the value is given as an extra parameter of intensity to the *rules* set (*cf.* class 5 in III-B).

A sign $+/-$ (no sign given means +) indicates if the operator implements the concept or its antonym.

B. Eight classes of operators

Influence operators can be applied to four kinds of operands: goals (γ), actions (α), plans (π) or functions (φ), depending on their class (*cf.* Table II). We distinguish eight main classes of operators:

1. Goal operators are used during Deliberation step to control the management of goals, *e.g.* to elicit goals from desires, to arrange goals in the goal stack *etc.*

2. Preference operators are used during Decision step, to compare and sort in total or partial order a set of rationally equivalent $\alpha|\pi$ (*e.g.* to sort S_p, S_o or S_d sets). Then, it becomes possible to implement mental attitudes such as likes and dislikes, in terms of minimal/maximal elements. In order to compare two items according to a given criterion, one needs a normalized measure function (or cost function) $\chi_{criterion} : \mathbb{F} \mapsto [0.,1.] \cup \{\perp\}$, which implements the criterion. The cost function is application dependent, meaning it has to be defined for each application; if not, it is not an error, it just maps to $\perp$ and is considered void. Hence to each preference operator $P_{criterion}$ is uniquely associated a cost function $\chi_{criterion}$. Comparing functions makes it straightforward to compare actions. By combining action-comparisons, it is often possible to compare plans[3].

For example, let three actions a_1, a_2 and a_3 be respectively linked to functions go-car, go-cycle and go-walk, and the application provides: $\chi_{easy}(\text{go-car}) = 1.$, $\chi_{easy}(\text{go-cycle}) = 0.5$, $\chi_{easy}(\text{go-walk}) = 0.1$. Then a 'lazy' agent could have a preference operator P_{easy} with a high level of activation ($v = 2$), making action a_1 to be the maximal element, whenever preferences are required (*e.g.* suppose $S_p = \langle a_1, a_2, a_3 \rangle$ then a_1 will be the psychologically preferred action in a set of rationally preferred actions — respectively for S_o or S_d sets). Note that preference-based operators have to deal with partial orders, and that maximal or minimal elements are not necessarily unique.

3. Filter operators are used to decide whether to keep or to discard an element $\gamma|\alpha|\pi$ for further considering in Deliberation or Decision steps. Filters define the *intrinsic* mental attitude of the agent towards $\gamma|\alpha|\pi$ (*i.e.* contrary to preferences, no comparison to other elements is performed). For example, if the level of filter operator F_{lawful} is > 0, then action a_1 linked to function kill-people will be discarded, even if it is executable and it was proposed as an efficient solution by the Planning step. Filters have precedence over preferences, *i.e.* preference operators decide among already intrinsically filtered elements.

4. Scheduling operators are used during Execution step to control the strategies of plan scheduling. They are divided into heuristic and alteration operators:

– Heuristic operators provide meta control over seq, alt, par, case operators in procedural plans, and over S_p, S_o or S_d sets in declarative plans. For example, if the level of scheduling operator $S_{predictable}$ is > 0 then $p_0 = par[a_1, a_2, a_3]$ will always be executed as $seq[a_1, a_2, a_3]$ (Note: executing a par as a seq is rationally correct).

– Alteration operators can alter the structure of procedural plan *e.g.* executing a seq as if it were a par is often (but not always) incorrect; it can be used in case of a very disorderly

[3]In all cases, comparisons are ultimately based on application-dependent information, meaning that two distinct agents elicit the same maximal/minimal elements. Hence our use of the term 'preference' is different from studies based on profiles (*e.g.* preferring red cars over blue cars). We do not consider profile-based preferences operators because they are traditionally integrated into the Planning step (as static external constraints) and they are not considered by psychologists working on personality traits.

agent. They can also alter the scheduling process itself *e.g.* remixing even deleting goals in the goal stack. Alteration operators can lead to non rational behaviors, therefore they are not discussed further.

5. Modal operators are used during Execution step to control the actual execution of the function linked to an action, in terms of close related notions of gradualness and intensity, which are provided by application functions. Modal operators can be mapped on plans, recursively down to actions.

– Intensity operators have an influence on functions with intensity *params*. For example, let a_2 be linked to function $\phi = \langle f_2, args, \{speed, \ldots\}\rangle$. If the level of the modal operator M_{quick} is $v = 2$ then action a_2 will be executed with maximum value for parameter *speed*.

– Gradual operators have an influence on functions handling gradual levels of achievement. Several cases of graduality can be considered. The simplest case is a function that can be achieved in a typical manner (like doing the 'par' in golf), more than the par, less than the par. For example, function give-beggar-money would be affected by the modal operator $M_{nowaste}$, leading it to be either: a) achieved in a typical manner (the 'par' being around 1$ bill for $M_{nowaste} = 0$); b) overdone (give a 100$ bill if < -1); c) underdone (give a 5 cents coin if > 1). Note that one has to give something otherwise the action would not be considered as executed.

6. Option operators are used during Execution step to add the execution of extra α after (respectively prior to) the execution of the current action. So they are prologue and epilogue add-ons that can modify the postconditions but must not alter the goals (respectively must not alter the preconditions for epilogues). For example, let action a_1 be linked to function open-door and a_2 to pass-door, making the plan $p_1 = a_1; a_2$; moreover, let action a_3 be linked to close-door. If the level of modal operator $O_{cleanup}$ is > 1, then action a_3 can be added as an epilogue to plan p_1. Prologue actions (like wash-hands before doing something) or epilogue actions are application dependent; again if not provided, they are just considered as void influences.

7. eXpectation operators produce expected facts during Deliberation and Decision steps in order to be further used during Evaluation step. They represent the attitude of the agent towards the expected outcomes of an intended γ in terms of its $\alpha|\pi$ postconditions. Three main kinds of mental attitudes can be defined: hope, fear and unconcern. For simplification, we take here the psychologically-strong position that an agent cannot have attitudes about necessary postconditions $post_N$ (*e.g.* like a stoicist, an agent cannot fear a necessary bad outcome). However agents can have attitudes about possible postconditions. Its attitudes partition the $post_P$ set of $\alpha|\pi$ into three subsets: hoped $post_H$, feared $post_F$ and unconcerned $post_U$, leading to the partition $post = post_N \cup post_H \cup post_F \cup post_U$. For example, taking again action a_0 linked to cast-dice, proposition $upface = 6$ can be both a goal and a hope ($upface = 6 \in post_H$ of a_0).

8. Appraisal operators are used during Evaluation step to control the impact upon the mental states of the agent of the comparison between the expected outcomes and the actual outcomes of the latest executed item.

IV. Trait implementation

A. Mental model of the agent

Typically, the mental states of a person can be classed according to two main criteria:

– *Dynamicity* While several levels of dynamicity of mental phenomena are possible, we will just consider two levels: static traits and dynamic feelings;

– *Arity* Traits and feelings can either be intrinsic to the agent viewed as an autonomous entity, or related to interpersonal relationships, the agent being considered as part of a community of agents.

These two criteria entail a generic classification of mental states in four types:

Features (Ψ_F) correspond to typical personality attributes that can be considered stable during the agent 'lifetime'.

Relations (Ψ_R) represent generic attitudes of the agent towards other actors in a given community. Note that relations must not be confused with *roles* that define social relations between agents (boss, father *etc.*), not addressed here.

Moods (Φ_m) represent intrinsic mental states of the agent that can evolve under the influence of a) external events in the world, b) evaluation of its own actions upon the world.

Affects (Φ_a) represent interpersonal attitudes of the agent that can evolve along its interaction with other agents. While relations are generic, affects are always associated with specific agents in the community.

Despite simplifications, this classification covers most significant notions discussed in mental states modeling [22].

B. Influence operators for FFM trait Conscientiousness

As mentioned in Section I-B, we rely on a resource where WordNet glosses and Goldberg's q-items are categorized in bipolar behavioral schemes positioned in the FFM/NEO PI-R taxonomy. An excerpt from the resource, illustrated in Figure 1, shows the two bipolar schemes associated with the first facet of trait Conscientiousness. Schemes are defined and presented in [29]. The total amount of elicited schemes is 69 *i.e.* 138 bipolar positions. The distribution of schemes over the FFM/NEO PI-R domain is quite even, with an average of 2.3 schemes by facet. It is interesting to note that the distribution of schemes over the intrinsic/interpersonal trait criterion is also quite even (Ψ_F = 46% and Ψ_R = 54%). Although Ψ_R is certainly worth considering, in this paper we focus on Ψ_F and let Ψ_R for further work.

Considering glosses/q-items contained in the schemes definition resource file, it is possible to exhibit influence operators associated with the described behaviors. In this

Table IV
LEVELS OF ACTIVATION OF THE 30 INFLUENCE OPERATORS SUPPORTING 12 SCHEMES OF FFM FACTOR CONSCIENTIOUSNESS.

Facets		Competence		Orderliness		Self-Discipline			Achievement -striving		Deliberation		Dutifulness part.	Dutifulness Ψ_R schemes (not treated)			
Schemes / Operators	Generic agent (default)	ASSURED INSECURE ⊕ ⦶	DECISIVE HESITANT ⊕ ⦶	METHODICAL DISORGANIZED ⊕ ⦶	TIDY MESSY ⊕ ⦶	CONCERNED UNCONCERNED ⊕ ⦶	ATTENTIVE INATTENTIVE ⊕ ⦶	MINDFUL FORGETFUL ⊕ ⦶	AMBITIOUS UNAMBITIOUS ⊕ ⦶	HARDWORKER LAZY ⊕ ⦶	STRATEGIC TACTICAL ⊕ ⦶	COMMONSENSE UNSOUND ⊕ ⦶	LAWABIDING LAWLESS ⊕ ⦶	LOYAL UNFAITHFUL ⊕ ⦶	VIRTUOUS IMMORAL ⊕ ⦶	TRUTHFUL UNTRUTHFUL ⊕ ⦶	FAIR PREJUDICED ⊕ ⦶
G_{forget}	1							0 2		0 2							
G_{quick}	1		2 -2							1 -2							
G_{deduct}	1										2 0						
$G_{prudent}$	1										2 0						
$G_{nowaste}$	1										2 0						
$G_{knowledge}$	1											2 0					
G_{reason}	1											2 0					
P_{easy}	1									0 -2							
P_{safe}	1	0 2															
P_{clean}	1				2 -1												
F_{lawful}	1												2 -1				
F_{safe}	1						2 0										
$F_{ambitious}$	1								2 0								
S_{order}	1			2 -1									2 0				
$S_{predictable}$	1			2 -1													
M_{focus}	1	2 -1	1 0	2 -1		2 0	2 0		2 0	2 -2							
$M_{precision}$	1		1 0			2 0	2 0		2 0	2 -2							
$M_{tenacity}$	1	2 -2	2 -1			2 0			2 -2	2 -1							
M_{quick}	1		1 -2							1 -2							
M_{dopar}	1			2 -2					0 2	2 -2							
$M_{nowaste}$	1			2 0													
M_{clean}	1				2 -1					1 -2			2 0				
O_{extra}	1			0 2						2 0							
O_{play}	1									0 0							
$O_{cleanup}$	1				2 0					2 0			2 -1				
$X_{success}$	1	2 -2							2 0								
X_{hope}	1	2 -2							2 0								
$X_{choiceConf}$	1	2 -2	2 -2														
$A_{responsible}$	1					2 0				2 -1							
$A_{success}$	1								2 -2	1 -1							

section, we take as an example the FFM trait Conscientiousness, dealing mostly with Ψ_F. It is composed of 6 NEO PI-R facets containing 16 schemes, as given in Table IV. In trait Conscientiousness, 11 schemes have influence operators $\in \Psi_F$, while 4 others $\in \Psi_R$ (LOYAL, VIRTUOUS, FAIR, TRUTHFUL) are not treated. Scheme LAWABIDING, having influence operators in both sets, is partially treated. A set of 30 influence operators required to support the 12 Ψ_F-related schemes of FFM trait Conscientiousness (called $\mathbb{I}_C$) are listed in Table III with their semantics given as a gloss.

Given a set of operators elicited from a trait, it is possible to define an implementation of its schemes in terms of activation levels (as defined in Section II-B) of these influence operators. Then, the levels of activation control the cycle of the rational agent defined in Section II-A, according to their class, defined in Section III. Activation levels are defined with two main principles:

– They are relative values, describing a positive/negative tendency/deviation from an average behavior supposed to describe a 'normal' person, further called the *generic agent*;
– They are defined in a so-called 'salience table' schemes × operators, where each cell contains by default the activation level of the generic agent except in cases where the glosses/q-items of the scheme specify a deviation from average behavior.

Generic agent: From a psychological viewpoint, this notion is still an open question, which exceeds this study. Here, we will take a restricted position where a generic agent is defined in terms of low levels of activation for the influence operators. A strict definition $\forall \omega_i, level(\omega_i) = 0$ is not correct because most people have some degree of confidence, orderliness, dutifulness *etc.* corresponding to a positive, low/medium activation level *i.e.* 1 (except for some traits like Neuroticism where it would be -1). When several

Conscientiousness **Trait**

Competence **Facet**

ASSUREDNESS **Scheme**

+ ASSURED **Positive pole**

366 : showing poise and confidence in your own worth **Wordnet gloss**
479 : having or marked by confidence or assurance
947 : marked by excessive confidence
1290 : marked by excessive complacency or self-satisfaction
Q64 : Am sure of my ground **Goldberg's item** (Qi)

- INSECURE **Antonym**

136 : lacking self-confidence
230 : showing fear and lack of confidence
680 : lacking or indicating lack of confidence or assurance

DECISIVENESS

+ DECISIVE

478 : persuaded of; very sure
477 : not liable to error in judgment or action
265 : characterized by decision and firmness
Q61 : Complete tasks successfully
Q62 : Excel in what I do
Q63 : Handle tasks smoothly
Q65 : Come up with good solutions
Q66 : Know how to get things done

- HESITANT

1106 : having or feeling no doubt or uncertainty; confident and assured
790 : unsettled in mind or opinion
791 : marked by or given to doubt
856 : lacking decisiveness of character; unable to act or decide quickly or firmly
1203 : uncertain how to act or proceed
Q67 : Misjudge situations
Q68 : Don't understand things
Q69 : Have little to contribute
Q70 : Don't see the consequences of things
Q100 : Postpone decisions

Figure 1. Excerpt from the Web resource on schemes in FFM/NEO PI-R

operators are to be applied in the same cycle step and they have the same level of activation, they must be prioritized to avoid contradictory influences. For a generic agent, a static pre-order on schemes can be predefined.

Example: To keep the example simple, we consider here a generic agent having a low level of activation for each operator in Table III: $\forall \omega_i \in \mathbb{I}_C$, $level(\omega_i) = 1$. The salience table of 12 Ψ_F-related schemes of Conscientiousness is given in Table IV where empty values = 1 and non empty values describe the deviation from generic agent's values.

Designing a specific character: Given Table IV, one can define a specific agent by selecting which schemes the agent is likely to exhibit. Typically, a person only exhibits few tendencies deviating from average behavior; hence in a human-like agent, few schemes will be selected (say two or three). Note that formally, it is possible to select an arbitrary number of schemes. When more than one scheme is selected, it can happen that for a given operator ω_i, two schemes induce conflicting levels (*e.g.* a TIDY and LAZY agent for M_{clean}). Similarly to the generic agent, it is possible to predefine a static pre-order on schemes. Moreover, when human-like agent are defined, conflict of operators can be a source of realism (*e.g.* by making obvious the agent's hesitation between two tendencies like TIDY and LAZY).

Table III
$\mathbb{I}_C$: OPERATOR SET FOR CONSCIENTIOUSNESS

Op.name	Gloss of + pole (informal semantics)
$G_{forget}+$	forgets γ (*e.g.* drops from goal stack)
G_{quick}	deliberates quickly (*e.g.* takes first in list of alternatives)
$G_{deduct}+$	controls the depth of deduction process
$G_{prudent}$	considers consequences of γ
$G_{nowaste}$	dislikes γ that waste resources
$G_{knowledge}+$	* has a large knowledge base (facts)
$G_{reason}+$	* has a large common sense base (heuristics)
P_{easy}	prefers easy $\gamma\|\pi$
P_{safe}	prefers safe $\gamma\|\pi$
P_{clean}	prefers clean $\gamma\|\pi$
F_{lawful}	discards unlawful $\gamma\|\alpha\|\pi$
F_{safe}	discards unsafe $\gamma\|\alpha\|\pi$
$F_{ambitious}$	prefer ambitious $\gamma\|\alpha\|\pi$
S_{order}	executes $\alpha\|\pi$ in order (- pole: seq⇒par)
$S_{predictable}$	executes $\alpha\|\pi$ routinely (*e.g.* par⇒seq)
M_{focus}	executes $\alpha\|\pi$ with mental concentration
$M_{precision}$	executes $\alpha\|\pi$ with precision
$M_{tenacity}$	perseveres if failures/difficulties during $\alpha\|\pi$
M_{quick}	executes $\alpha\|\pi$ quickly
M_{dopar}	control graduality of execution (see II- 5 Modal/gradual)
$M_{nowaste}$	don't waste resources while executing $\alpha\|\pi$
M_{clean}	is clean while executing $\alpha\|\pi$
$O_{extra}+$	controls general tendency to execute optional α from S_o
O_{play}	adds pleasurable α, taken from S_o
$O_{cleanup}$	adds clean-up action(s) (not in S_o) after execution of $\alpha\|\pi$
$X_{success}$	expects success of current $\gamma\|\alpha\|\pi$
X_{hope}	has high level of hopes in $poss(\alpha\|\pi)$
$X_{choiceConf}$	confident in its choices of $\gamma\|\alpha\|\pi$
$A_{responsible}$	feels responsible for the results of its $\gamma\|\alpha\|\pi$
$A_{success}$	has a particular sensitivity to success

* operator overlaps with capacities in rational reasoning.
\+ negative level values don't apply for these operators.

V. CONCLUSION AND FUTURE WORKS

We have presented an approach based on the principle that personality traits a) can influence the deliberation cycle of rational agents and b) can be defined in terms of level of activation of influence operators. A set of eight classes of influence operators has been described and it has been applied to one of the five main classes of the FFM taxonomy: Conscientiousness, which can be expressed as a set of 30 operators, belonging to the eight classes.

In this study, we have focused on intrinsic personality traits (Ψ_F), which cover only 46% of the behavioral schemes. Remaining schemes are related to interpersonal relationships (Ψ_R), which require a deliberation cycle adapted to interaction. In further work, we intend to extend our model to handle Ψ_R-based traits, following the same approach. Another development will be the study of characters containing complex characters entailing conflicting influences over the rational process.

References

[1] G. W. Allport. *Personality: A psychological interpretation*. New York: Holt, 1937.

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

[3] F. Bouchet and J. P. Sansonnet. Classification of wordnet personality adjectives in the neo pi-r taxonomy. In *Proc. of the Fourth Workshop on Animated Conversational Agents, WACA 2010*, Lille, France, 2010.

[4] Michael E Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, 1987.

[5] Justine Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Villiamsson, and H. Yan. Embodiment in conversational interfaces: Rea. In *CHI'99: Proceedings of the SIGCHI conf. on Human factors in comp. syst.*, pages 520–527, New York, NY, USA, 1999. ACM Press.

[6] R. B. Cattell, H. W. Eber, and M. M. Tatsuoka. *Handbook for the sixteen personality factor questionnaire (16 PF)*. Champain, Illinois, 1970.

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

[8] Mehdi Dastani. 2APL: a practical agent programming language. In *AAMAS '08:The seventh international joint conference on Autonomous agents and multiagent systems*, volume 16, pages 214–248, Estoril, Portugal, 2008. Springer-Verlag.

[9] Tiago Doce, Joao Dias, Rui Prada, and Ana Paiva. Creating individual agents through personality traits. In *Intelligent Virtual Agents (IVA 2010)*, volume 6356 of *LNAI*, pages 257–264, Philadelphia, PA, 2010. Springer-Verlag.

[10] Rick Evertsz, Franck E. Ritter, Paolo Busetta, and Matteo Pedrotti. Realistic behaviour variation in a BDI-based cognitive architecture. In *Proc. of SimTecT'08*, Melbourne, Australia, 2008.

[11] Sybil B. G. Eysenck, Hans Jorgen Eysenck, and Paul Barret. A revised version of the psychoticism scale. *Journal of Personality and Individual Differences*, 6(1):21–29, 1985.

[12] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

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

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

[15] B. Hayes-Roth, L. Brownston, and R. van Gent. Multiagent collaboration in directed improvisation. In *Proc. of the First Int. Conf. on Multi-Agent Systems ICMAS 95*, pages 148–154, San Francisco, CA, 1995.

[16] Nick Howden, Ralph Rnnquist, Andrew Hodgson, and Andrew Lucas. Intelligent agents - summary of an agent infrastructure. In *Proc. of the 5th International Conference on Autonomous Agents*, Montreal, 2001.

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

[18] Mei Lim, Joao Dias, Ruth Aylett, and Ana Paiva. Improving adaptiveness in autonomous characters. In Helmut Prendinger, James Lester, and Mitsuru Ishizuka, editors, *Intelligent Virtual Agents*, volume 5208 of *LNCS*, pages 348–355. Springer Berlin / Heidelberg, 2008.

[19] Pattie Maes. Artificial life meets entertainment: lifelike autonomous agents. *Commun. ACM*, 38(11):108–114, 1995.

[20] Lori Malatesta, George Caridakis, Amaryllis Raouzaiou, and Kostas Karpouzis. Agent personality traits in virtual environments based on appraisal theory predictions. In *Artificial and Ambient Intelligence, Language, Speech and Gesture for Expressive Characters, AISB 07:*, Newcastle, UK, 2007.

[21] M. McRorie, I. Sneddon, E. de Sevin, E. Bevacqua, and C. Pelachaud. A model of personality and emotional traits. In *Intelligent Virtual Agents (IVA 2009)*, volume 5773 of *LNAI*, pages 27–33, Amsterdam, NL, 2009. Springer-Verlag.

[22] Andrew Ortony. On making believable emotional agents believable. In Robert Trappl, Paolo Petta, and Sabine Payr, editors, *Emotions in humans and artifacts*, pages 189–211. MIT Press, Cambridge, MA, 2003.

[23] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The Cognitive Structure of Emotions*. Cambridge, UK, cambridge university press edition, 1988.

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

[25] Anand S. Rao and Michael P. Georgeff. BDI agents: From theory to practice. In *Proc. First Int. Conference on Multiagent Systems (ICMAS-95)*, pages 312–319, 1995.

[26] P. Rizzo, M. V. Veloso, M. Miceli, and A. Cesta. Personality-driven social behaviors in believable agents. In *AAAI Symposium on Socially Intelligent Agents*, pages 109–114, 1997.

[27] D. Rousseau. Personality in computer characters. In *AAAI Technical Report WS-96-03*, pages 38–43, 1996.

[28] N. Sabouret and J. P. Sansonnet. Learning collective behavior from local interactions. In *CEEMAS: Second International Workshop of Central and Eastern Europe on Multi-Agent Systems*, volume 2296 of *LNCS*, pages 273–282, Cracovia, Poland, 2002. Springer Verlag.

[29] J. P. Sansonnet and F. Bouchet. Extraction of agent psychological behaviors from glosses of wordnet personality adjectives. In *Proc. of the 8th European Workshop on Multi-Agent Systems (EUMAS'10)*, Paris, France, 2010.

[30] Sebastian Sardina. A course on BDI architectures. http://goanna.cs.rmit.edu.au/~ssardina/courses/Roma07PhDcourse/, 2007.

[31] B. G. Silverman, M. Cornwell, and K. O'Brien. Human behavior models for agents in simulators and games: Part I: enabling science with PMFserv. In *PRESENCE: Teleoperators and Virtual Environments*, volume 15, pages 139–162, 2006.