

A Flexible Framework for Representing Personality in Agents

André M. C. Campos¹, Emanuel B. Santos¹, Anne M. P. Canuto¹, Rodrigo G. Soares¹,
João Carlos Alchieri²

¹ *Informatics and Applied Mathematics Department, Federal University of RN Natal, RN - BRAZIL, 59072-970
Telephone: +55-84-3215-3813*

² *Psychology Department, Federal University of RN Natal, RN - BRAZIL, 59072-970
Telephone: +55-84-3215-3590*

{andre, anne}@dimap.ufrn.br, {emanuel, rodrigo}@ppgsc.ufrn.br, jcalchieri@brturbo.com

ABSTRACT

Aiming to represent individuals in a very realistic way, several works have attempted to introduce personality characteristics into artificial agents, usually based on theories of human personality. However, different theories require different structures in the agent architecture, reflecting on the way they influence agent behavior. This paper presents a flexible framework to generalize the influence of personality traits on the behavior of the agents. The framework provides an easy way to test and compare different personality models. In order to validate the framework, a simulation of working environment was developed and three personality theories (Big Five, Jung and Millon) were tested.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Intelligent agents, Languages and structures, Multiagent systems. I.6 [Simulation and Modeling]: Applications.

General Terms

Design, Standardization.

Keywords

Agent architecture, Agent-based framework, Personality-based agents.

1. INTRODUCTION

The notion of autonomy is an essential concept for social systems. From the social point of view, autonomy is an important aspect to distinguish individuals within a society. This concept is largely used in the agent paradigm when modeling social systems [6].

When modeling human social systems, autonomy becomes intrinsically related to how the agents behave and what make

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. to copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06, May 8-12, 2006, Hakodate, Hokkaido, Japan.

Copyright 2006 ACM 1-59593-303-4/06/0005...\$5.00.

them to behave differently for each other. In other words, it is necessary to identify what makes two individuals playing the same roles in a society (or in a group) and facing the same situations to behave differently.

Psychologists have tackled this issue for several decades through what they named *Personality* [2]. Personality represents the structured and dynamic set of characteristics of an individual, normally acquired from the environment and personal experiences [4]. The study of personality aims the understanding of the dimensions that allow individuals to be psychologically distinguished from another [13].

Therefore, for social systems, autonomy of an agent is also a result from its personality. In order to take the autonomy of agents into account when modeling human social systems in a low-level of abstraction, it is necessary to include the notion of individual personality. Several agent actions and characteristics, even the internal ones, may emerge from the encounter of the event perceived by the agent and its internal personality.

The use of the multi-agent paradigm for building social system simulations leads to a more natural way to represents an observed system, since its dynamics is expressed in terms of interactions between autonomous agents. However, most of the agent architecture used in multi-agent simulations considers the agent behavior based only on rational processes (e.g. BDI [5]). This does not provide tools for representing the chaotic nature of social systems. The use of personality in multi-agent applications for simulating social systems can lead to more chaotic (and realistic) results. This is one of the objectives of works dealing with synthetic actors with personality [8][12][10] and also in crowd simulations [7].

Another example of social scenario where the personality of the individuals has strong influence on the emergent behavior is in a working environment [14]. Individuals playing the same role in a working organization may have very different reaction (and consequently different performances in their tasks). This happens, among other reasons, because they have different personalities, affecting, as a consequence, the behaviors of these individuals.

There are several psychological theories and paradigms highlighting the concrete aspects of the influence of personality on the way in which we behave. However, none of them has reached a complete consensus among psychologists. Indeed, personality is subjective and subject to different interpretations. Specifically for working environment, in the Organizational

Psychology area, several models have already been proposed. Different companies may use different approaches for understanding their employees. A simulation application aiming to be generic enough to be able to represent different models will certainly face the intrinsic characteristics that is present in a model, but not in another one.

In order to solve this issue, it is necessary to conceive an agent architecture able to be adaptable to different models of human personality. Surrounding this architecture, a framework must be placed for providing enough flexibility to take into account several facets of the human personality. This paper describes the framework developed to smooth out the aforementioned problem. Flexibility is provided by description files and action scripts, separating the agent structure from its personality and behavior. Three models were implemented for testing purposes, based on the theories of Big Five [3], Jung [9] and Millon [11]. Also, a simulation application, named SimOrg – Simulation of Human Organizations, was adapted to this framework. This paper presents the framework structure and how personality-based behaviors are modeled in it.

Before presenting how different personality models can be handled in the proposed framework, it is necessary to analyze some characteristics usually found in different theory models. The Section 2 presents three theories already used in previous works related to agents. Models of those theories are also used for validating the framework. With those concepts in mind, it is possible to more easily understand the choices for the framework, which is presented in the following section (Section 3). In this section, it is pointed out how to generically represent personality in agents. The generic framework for personality-based agents is described in Section 4. In this section, it is also presented a tool to generate the scripts and description files for the framework. After, it is presented an application used to validate the framework. The application is a simulation tool for project management.

2. HUMAN PERSONALITY

Among the various paradigms (psychoanalytic, behaviorist, humanistic, biology and evolutionary, and personality traits), personality trait is the most widely accepted theory. This approach structures the human personality by a hierarchical model of traits. The behavior of an individual might be “interpreted” as a combination of the different levels of intensity of the different traits. The next sections present three different approaches for modeling personality traits. They will be used to validate the framework proposed in this paper.

2.1 The Big-Five Model

Among the several ways of organizing personality traits, the “Big-Five” dimensions (used by the Five-Factor Model [3]) have a large acceptance by psychologists, as well as computer scientists modeling personality into human-like artificial characters (virtual humans) (see, for instance [6]). It describes the human characteristics according to five main factors: extraversion, agreeableness, conscientiousness, openness to experience and neuroticism.

The combination of those five factors reflects the several aspects of human personality. For instance, an authoritarian person would be high on conscientiousness, but low on agreeableness and openness to experience [1] (example cited in [4]). Other combinations might be possible, resulting in a large variety of personality styles.

Although it is widely accepted, the Big-Five Model presents a constraint for us, who are modeling artificial behaviors (and not

studying people). Whereas the Big-Five Model maps a behavior into its five factors, one cannot derive a behavior from the level of the factors. In other words, it is not possible to predict if someone is authoritarian only by the means of the level of his/her conscientiousness, agreeableness and openness to experience. Individuals with the same level of those factors might behave differently for the same situation. Indeed, a behavior is also a function of the personal experiences from the past and the personal ambitions for the future.

2.2 The Jung Model

Carl Gustav Jung has also proposed a theory that attempted to describe the differences of personality using categories of introversion and extroversion and also using cognitive and emotional functions. These aspects can be seen as the basic dimensions of the individual differences. According to Jung, these dimensions offer a basic typology needed to classify individuals [7].

In order to identify a psychological type, it is necessary to determine the fundamental attitudes of an individual. In other words, to define whether a person is preferentially guided through its internal world (introversion) or through its external reality (extroversion). Once it is defined, an individual can be classified according to its dominant function, also known as *Main Function*, which describes the preferential behavior of the individuals. There are four possible psychological functions that can be chosen as dominant, which are the following: Thought, Feeling, Sensation and Intuition.

Jung divided the four aforementioned functions into two classes, rational and irrational. The rational functions (thought and feeling) help to make judgments and decisions, and the irrational functions (sensation and intuition) supply information in which it can be based on those judgments. The fundamentals attitudes (introversion and extroversion) may be combined with the four functions (thought, feeling, sensation or intuition), which can lead to eight different combinations, constituting eight psychological types to represent the dominant function of an individual. In order to provide a more complete description of an individual, a classification of the least preferred behavior can also be defined. It uses the same function classification of the dominant function and it is called auxiliary function. It defines the function that an individual utilizes to cope with the direction least preferred situations.

According to Jung, all the aforementioned functions are present in every individual, in higher or smaller degree, but just one is predominant, which is defined in the *Main Function* [3]. The auxiliary function is identified as the function that someone uses to work with less favorite directions and it is usually the opposite of the main function. For example, if the main function is one of the rational functions (thought or feeling), the marginal function it will be one of the irrational functions (sensation or intuition) or vice-versa [3].

2.3 The Millon Model

The theoretical model of personality proposed by Millon takes into account the importance of biological factors such as genetics [11]. These factors are described in one specific matrix for each individual, which represents a fundamental role in the formation of personality [13]. This matrix is also composed of environmental perceptions and actions taken during conflict situations. During the development of an individual, personality has influence from biological and psychological factors, that interact in an endless spiral, in which each circle of this spiral is constructed over previous interactions, creating, in this way, new bases for the next interactions [14].

Millon proposed a measurement to express personality, which is based on the theoretical comprehension of the actions taken to reach the goals that an individual has in his/her life. Also, it takes into account the way to process information received from the environment. In this sense, it has been elaborated a tool to verify a dynamic configuration of interactions, representing three large areas: motivational aims, cognitive styles and inter-personal relations.

According to Millon, personality can be viewed as a result of twelve bipolar¹ attributes. The area Motivational aims includes three bipolarities, linking the ecological and evolutionist theories. It is based on the conceptual antecessors of these two theories through three main formulations: the existence of organisms, its adaptation to the environment and the answers provided in this relation. Based on this theoretical model, Millon presented the following polarities: openness versus preservation, modification versus accommodation and individualism versus protection. In these groups of attributes, it can be observed the orientation of an individual in relation to the role of the environment as a motivation of the actions of an individual [14].

The cognitive styles can be found in the evolutionist perspective as well as in contribution of authors such as Jung and Myers. They are related to how individuals are oriented when interacting to the environment. Their main aim is to evaluate the way an individual can process information, along with three main models proposed by Jung (extroversion versus introversion, feeling versus thinking and sensation versus intuition). Based on this and in [15], the bipolarities can be defined as extroversion versus introversion, feeling versus thinking, reflection versus affectivity and systematization versus innovation.

Millon proposed to use the inter-personal component to evaluate the style of the relationship of an individual with the others. All bipolarities of this area are based on a bi-psychosocial theory of reinforcement learning as well as active and passive strategies. The bipolarities are: shyness versus communicability, doubt versus security, discrepancy versus conformity, control versus submission and satisfaction versus dissatisfaction.

3. REPRESENTING PERSONALITY INTO AGENTS

One of the issues of representing personality into agents is the transposition of the concepts coming from Psychology, in general difficult to generalize (each individual is unique), to the computational form, which functioning tries to systematize and, as a consequence, to generalize behavior. However, although personality is defined as “*an individual’s characteristics pattern of thought, emotion, and behavior*” [4], the main issue of representing human personality into agents is the use of the aforementioned theories on the inverse way in which there were conceived. Indeed, human-personality theories tries to generalize the behavior of an individual within a pattern group, but this does not necessarily means that knowing the group an individual bellow one can predict the individual behavior for every situation.

¹ Bipolarity can be seen as a complementary measure between two concepts. For instance, if an individual has 30% of modification, it automatically has 70% of accommodation.

In other words, from an observed behavior pattern, it is possible to categorize an individual (according to a theory), however it is hard to generalize a behavior from a categorized individual.

The models used by psychologists to acquire personality traits of an individual aim to “classify” the individual behavior pattern according to the different dimensions (for instance, the five-factor model deals with five characteristics). This classification is then a function of the observed behavior pattern (or from a pattern acquired from filling a questionnaire), mapping the behavior to a classification. However, the mapping function is not bi-directional for every situation. An individual classified as introverted does not necessarily acts introverted for all situations, but at least in those ones observed or asked in her/his questionnaire.

In order to simplify this issue, without losing generality, the function that maps personality to behavior patterns can have an error parameter for representing possible variations on the behavior pattern. The model proposed by Jung attempted to solve this issue through the Auxiliary Function, which represents the behavior pattern used when the Main Function is not triggered (Jung worked deeply on the subject of unconsciousness). In the same way, an agent architecture dealing with personality must handle unpredictable behaviors of the agents. The agent behavior is then a function (Eq. 1) of the situation it is facing (s), its personality characteristics (p), and the level of error (e), which model the variation of the personality attributes in the agent behavior.

$$b = f(s, p, e) \quad \text{Eq.1}$$

The way in which the personality is represented and how it is handled, providing a general framework for personality-based agent systems, is presented in the next section.

4. A GENERAL FRAMEWORK

As previously mentioned, autonomy is an attribute intrinsically related to social systems. In the context of agent paradigm, this is related to Multi-agent systems, where agents are organized in one or more groups. Agents also perform tasks and, eventually contribute to each other through coordination mechanisms. The proposed framework takes those characteristics into account, conceiving a way to structure the agents inside its multi-agent system as well as how they coordinate their actions. The source of inspiration for defining the framework was the way in which individuals work within of a human organization. Several pieces of code were also adapted from a previous framework developed by one the author of this paper; the framework MAVIS [17]. The MAVIS framework was developed for ecosystem simulations based on reactive agents. The simulation structure of the framework (simulation kernel, event treatment, etc.) was conserved, but the agent-side of the framework was remodeled for taking into account the agent architecture presented in the paper.

In a working environment, individuals are grouped and structured in a hierarchical way, composing the configuration of organization. Individuals may also be member of several groups. This approach is similar to organizational model AALAADIN [15] used by the MadKit platform [16]. Usually, in working environment, there is also a process guiding the actions of individuals. The process indicates which activities must be performed for the first place, which are the dependencies of each activity, and so on. This idea was also translated to the framework proposed in this paper.

Then, the framework takes into account three different aspects of a multi-agent system:

- **The organizational structure:** that defines the static component of the system. In this model, the hierarchical distribution of agents in groups and the roles they will play inside their groups are defined. The roles are useful for determining the actions an agent is responsible for;
- **The organizational process:** that defines the dynamic component of the system. It structures how the agents integrate their actions to achieve a common goal. The organizational process is organized in activities, where each activity is an set of actions; and;
- **The behavioral component:** that defines how the agents perform the tasks described in the organizational process. It is in this component that personality attributes plays an important role. Each action that an agent performs will suffer influence on its personality attributes.

In order to provide flexibility for these three aspects, the framework takes in charge three description files, each one for a different aspect. In this sense, the organizational structure, process and behavioral component can be customized according to the application without changing the framework code.

4.1 The agent architecture

Each agent in the framework represents an individual within a multi-agent organization. The interactions, rules, structures and interaction protocols that happen within the organization are dependent to the way the agents process information. Then, it is necessary to model the internal architecture of the agents in order to structure the framework. Moreover, the agent architecture must also take into account personality attributes when enabling, disabling or activating an agent action.

The architecture for the agent is shown in Figure 1. As it can be observed, the defined architecture contains not only modules commonly used in a common agent architecture, such as perception, planning, goal definition and action, but also it is composed of a base, defining the artificial psychological profile of the agent.

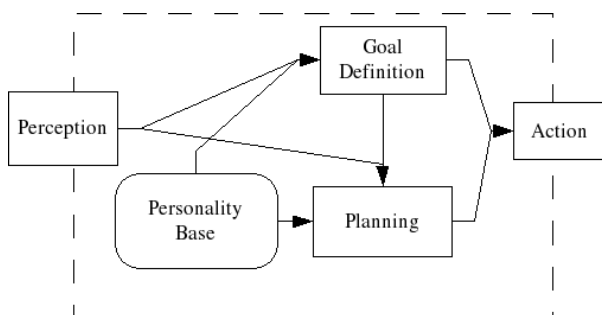


Figure 1 – The agent architecture.

As personality can be seen as the way in which an individual interacts with the environment, it is represented as an information base, or personality base. Depending on the personality model used by the framework in a specific application, the attributes used in that base might differ. For instance, for the model proposed by Millon, it is composed of the three aforementioned topics of the human personality (motivational aims, cognitive modes and inter-personal relations), each area having its respective bipolar attributes.

The personality base has strong influence in the definition of the goals of an agent (goal definition module) as well as in the choice of the actions to perform the chosen goal (planning module). In other words, when the module of Goal Definition is activated, it can request information from the personality base in order to define the agent goal.

In considering the personality base as a common knowledge base, this agent can be seen as a common BDI (Beliefs, desires and intentions), in which based on its beliefs, an agent can choose its goals as well as the needed actions to reach this goal. The use of personality will help to transform the modeling in a more natural one, in which the personality of an individual should and will be taken into account.

As it can be observed in Figure 1, there are some arrows indicating the influence of the personality base over the modules of Goal Definition and Planning. The arrows represent mapping functions from the personality to the respective modules. This mapping is provided by behavioral scripts, explained more in detail in next section.

In order to model the complex set of behaviors that an agent might perform within an organization, it is necessary a huge amount of rules associated to the agent. Depending on the role the agent plays inside the organization (or inside the groups of the organization), the agent must organize its activities (or tasks) according to a priority criterion. This priority order of the activities that an agent is responsible for is modeled by a horizontal layered architecture, similar to the reactive architecture defined by Brooks (Subsumption). This architecture was however extended to provide the following needs:

- **The ability to deal with two level of action abstraction:** The first level is composed of activities, which represents the tasks the agent are responsible for. For instance, an agent that is member of two groups might play different roles at the same time, having, as a consequence, to execute different activities. The second level is composed of actions. In fact, an activity is defined as a set of actions that might be performed in a specific sequence or not;
- **The ability to dynamically define priorities for activities:** In the agent architecture, the layers define the level of priority of an activity. However, this priority might be changed dynamically as a result of an action performed by the agent. Whenever the priority of an activity is changed, the respective set of actions assigned to the agent will also change.
- **The actions of the activity with the greater priority compete between themselves to be activated:** If an agent is executing an activity composed of several actions that one does not depends on the other, the independent action will compete between themselves to be activated, according to a probability degree. This degree is defined as a function that maps the attributes specified in the personality base and the corresponding action. This approach is suitable for representing the “personal” choice of the action, since there is no dependence of the competing actions.

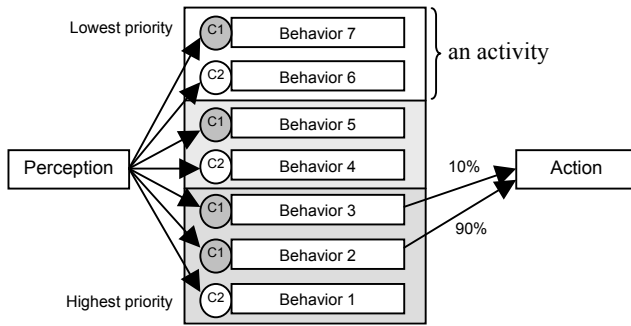


Figure 2 – Functional schema for the agent architecture.

The Figure 2 illustrates the functioning of the layered architecture. Each behavior rule, in order to be activated, needs that its antecessors (represented in the figure as C1, C2, etc.) be validated. Among the behavior rules of the activity with the highest priority (in the figure, represented by the Behaviors 1, 2 and 3), just one will be activated. Supposing that two of them have their corresponding conditions validated (C1), they will compete according to the “personal” interested of the agent. The system will then define the degree of probability for activating the behaviors (in the case illustrated by Figure 2, it is the behaviors 2 and 3) according to the personality attributes saved in the personality base, resulting in the most probable action the agent will take.

The way in which the agent architecture maps the personality to the actions is through a function, scripted when implementing a behavior action. The function itself is based on empirical feedback given by an expert (psychologist). There is no good or bad match between personality and actions, however it normally requires several simulation executions before the simulation model (the script function mapping personality and actions) be considered calibrated by an expert. The next subsection describes the simulation model more in detail.

4.2 Scripting Personality and Behavior

The personality model in the proposed architecture influences the agent behavior. Since, as previously mentioned, there are several ways to represent personality, it is necessary to adapt one of them in a specific application. In order to do this adaptation in easy way, we used a rapid prototyping strategy. We moved the behavior part that relates the actions execution and the personality models into a bean shell script. Therefore, in order to update the agent behavior, it is just necessary to alter the script file.

An action is a Java class that encapsulates an action script. The action accesses the script interpreter and, then, calls its methods separately. Each action method is related to a script method. The main methods that the script must implement in order to integrate with the framework are:

- **The evaluating method:** `validate()`: it is used for calculate the probability of the action to be executed, according to the personality parameters;
- **The quality method:** `getQuality()`: it calculates the amount of work the agent will do in that action. The higher the quality is, the better the action execution, and;
- **The execution method:** `execute()`: it performs the sequence of steps for an specific action.

Besides the methods, the script has the personality type data. The simulation uses that data to link the agent personality to the action ideal characteristics, influencing, in this way, the evaluation and quality methods. In other words, in order to have

a better chance to execute an action, the agent must have a personality that is close to the ideal profile described in that action.

As presented in section 3 (Eq.1), the behavior of an agent also takes into account a level error, representing unpredicted behavior. This error is a randomly generated value in $[-0.1;0.1]$ that is added to the personality attributes when the `validate()` and `getQuality()` methods are requested. Then, a level of error is introduced for simulating unpredicted behavior.

In order to facilitate the end-users of the framework, a tool for generating the scripts files was developed. As previously mentioned in the framework description, the framework deals with three different aspects of a multi-agent system. The tool attempts to facilitate the handling of theses aspects by generating three different files, all of them specified in XML format:

- The organization structure file.
- The work process file.
- Several files containing the personality and scripts.

The file containing the personality descriptors is written for each personality model. It consists on three main tags:

- The `script` tag containing pieces of bean shell script;
- The `personality_descriptors` tag describing the personality stereotypes for each model;
- And the `action_descriptors` tag describing the personality ideal profile to execute the action;

Figure 3 illustrates description of a personality following the Jung approach of personality. Bean shell pieces of code constitute the `script` tag in the XML file. These pieces of code compose the actions files. They have a method-like format to enable the action script generation. This kind of format is necessary to get the action to execute correctly for each model. Since the Jung, Big Five and Millon models have different approaches, it is not possible to use the same evaluation model for them.

```
<model theory="JUNG">
  <script>
    <method name="execute">
      int p =
      entity.getActivityPriority(activity.getName());
      runningTime++;
      aux = getQuality(entity);
      ...
    </method>
    <method name="quality">
      personality = entity.getPersonality().getName();
      a = ideal{personality};
      ...
      return new Double(a*p);
    </method>
  </script>
  <personality_descriptors>
    <personality name="ES">
      ...
    </personality>
  </personality_descriptors>
  <action_descriptors>
    <action name="code" size="700">
      <property name="ES" value="0.7"/>
    </action>
  </action_descriptors>
</model>
```

Figure 3 –XML file illustrating the personality description.

The `personality_descriptors` tag consists of personality stereotypes. Each stereotype is a personality pre-defined description. They were conceived by a specialist to give a theoretical basis to the use of human personality in the simulation.

Finally, the `action_descriptors` is composed of personality descriptors. This part of the XML file links the action to the personality models. This part of the file contains information about the ideal personality profile to execute the action, this data are used to generate the script file as well.

An agent action is composed of a sequence of several steps. The agent might read its messages in the mailbox update its internal states and/or execute its behavior pattern. Part of agent cognitive mechanism stay in the behavior. The behavior evaluates the list of activities in which the agent is involved. When an activity is chosen (by its priority), the agent create a list of actions. In next step, the agent validates its list accessing the method "validate" of action. This method answer if the action can be executed (it is the condition explained in Figure 2). When more that one action is valid, the agent begins the decision process. On that moment the behavior access the `evaluate()` method of the action. This method determine the probability of the agent execute an action. Then, the personality descriptor of the agent is compared to the personality descriptors of the action. The comparison result is used for provide the degree of probability that the action will be executed by the agent. Finally, when an action is chosen, it starts the its execution. To execute an action the method `execute()` of action is called. During the action execution, it is necessary to access the method `quality()`. This method, as well as, method `evaluate()` compare personality descriptor, but for a different purpose. The method `quality()` use the information of personality to determine the agent performance when execute the action. The greater is the quality of action performed by an agent better is the result from that execution. An example of personality profile used in the framework is illustrated in Figure 4.

```
<personality name="competitive">
  <!-- motivation targets -->
  <property name="openness" value="0.3"/>
  <property name="modification" value="0.7"/>
  <property name="protection" value="0.5"/>
  <!-- cognition mode -->
  <property name="extroversion" value="0.7"/>
  <property name="sensation" value="0.7"/>
  <property name="reflexion" value="0.5"/>
  <property name="innovation" value="0.1"/>
  <!-- interpersonal relationship -->
  <property name="communication" value="0.5"/>
  <property name="selfconfidence" value="0.5"/>
  <property name="discrepancy" value="0.3"/>
  <property name="submission" value="0.5"/>
  <property name="concordance" value="0.5"/>
</personality>
```

Figure 4 – Example stereotype for competitive agents.

The XML files can be written manually or using the graphical tool showed in Figure 5. The graphical tool can read, edit and write the XML files used by the framework. It is a friendly user interface, that it abstracts the user of the XML complexity.

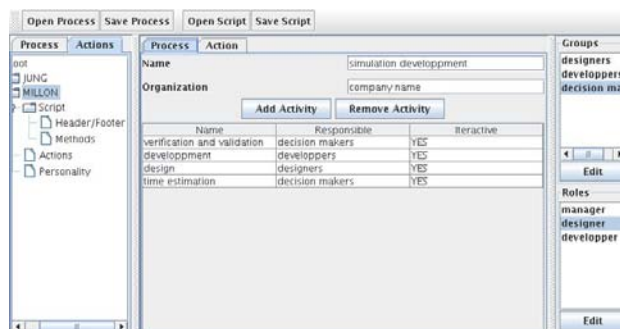


Figure 5 – Initial display of the XML generator tool.

5. APPLICATION TO WORKING ENVIRONMENT SIMULATIONS

The proposed framework was developed to solve the flexibility issue faced during the development of a simulation application, named SimOrg. This application aims to reproduce the organizational dynamics found in working environment and can be used for studying individual behavior within an organizational as well for education and training purposes.

As the original version of the SimOrg application does not give flexibility for representing personality-based agents on different models, a new version of the simulator was proposed in order to generalize the application. The adaptation of SimOrg to the framework provided the ability to accept a larger number of agent personality models such as the Millon, Jung and Big Five models.

5.1 The SimOrg Scenario

In order to analyze the influence of personality in the performance of the SimOrg agents, it has been defined a scenario for a working organization, where agents could not only interact and act according to their own personality, but also follow a working process (organizational rules). The scenario was based on the experience of the authors in developing software. In this sense, the fictitious organization is a software development company.

In the scenario, the chosen development process is an iterative one based on agile methodologies [18], which was simplified for testing purposes. Agile methodologies are ideal for testing the proposed architecture as individuals and their informal communication are very important for them. In other words, the success of a project is not only a matter of passing formal documents through staff., but also individual interactions that can represent a successful project.

The scenario was then represented as having formal, informal interactions and task actions, like *manager*, *design*, *estimate time*, *code*, *test code* and *validate design*. Those actions were organized into four activities, each one having an individual or group responsible for.

The communication acts follows a small subset of an ACL structure, composed of *performative*, *sender*, *receiver* and *contents*. Although this is enough for representing communication between agents, it was observed that human communication is not necessarily binary (to send or not to send a message). In order to model working human organizations, it is also necessary to represent how good (or bad) an individual performs an action, including communication acts, and the importance that an individual gives for those actions. For instance, suppose that a very introspective individual can not perform a task as desirable. Although there is reluctance in

asking for help, s/he will end up asking for a colleague help. The way in which s/he performs such an action (*quality*) and the need for doing that (*importance*) will certainly be influenced by his/her personality (very introspective). As a consequence, we have introduced in our ACL subset two other components: *quality* and *importance*, whose values are functions of the individual personality attributes (openness, introversion etc.). In this way, an agent is also able to react to a received message according to its quality and importance.

5.2 Scripting the scenario

The structure of the software company described in the scenario is composed of managers, designers and developers; these are meant as roles in the organization and on the simulation as well. The workers (agents) are allocated on groups that are semantically separated according to the roles of their agents. An agent performs one or many of those roles in the organization depending on the groups in which the agent is. In other words, for each group that the agent A belongs, a role is associated to A in the group. A group must always have roles agents that will play them. In this XML script, this is represented by the organization description file. The Figure 6 shows a piece of the XML file that describes the agents and its roles in a group.

```
<group name="X" type="developpers">
  <agent-role agent="C" role="manager"/>
  <agent-role agent="E" role="developer"/>
  <agent-role agent="F" role="developer"/>
</group>
```

Figure 6 – A piece of code representing a group and the roles and agents associated to it.

Once the static part of the organization is defined, the agents will have to perform some actions. The organization sets the actions that each role will execute, so the agent will perform all the actions associated to its roles like in real organizations. The set of possible roles is: Manager, Designer and Developer. And the set of actions is: Manager, Estimate Time, Design, Code, Validate and Verify.

The organization, in order to accomplish the software product manufacturing, must complete some activities in certain order. An activity is a part of the organization's process; it consists of an actions sequence, so when all the actions in one activities are done, the activity itself will be over. Activities can depend on other activities in a certain percentage, that is, suppose that the activity B depends on the activity A in 30%, then when 30% of A is done, B can begin. There are four activities in the organization process: time estimation, design, development and verification and validation. The design activity depends on time estimation, the development activity depends on 30% of the design and verification and validation activity depends on 30% of the development activity.

The actions were conceived in order to clearly distinguish the agent structure: its motivation, fatigue, roles; the agent personality and the action itself. For each action is defined in its script an ideal suitable personality to perform that action, the closer the agent personality is to that ideal, the more efficient the agent will be executing that action. Each agent has an attribute called motivation, this feature also defines the quality of the agent work on the action, if an agent has got a low motivation, this will affect negatively in the action performed by that agent. Another attribute is the fatigue; it influences the agent motivation, so indirectly influences the quality of the agent work.

All those settings for the organization and the process are made in the XML file. The user can share the scenario file that he has

build in the application through the Web and then other users can construct their simulation scenario based on that one.

5.3 SimOrg Application

At this stage of the implementation, a stable version of the Simulator is ready and on use. Some tests have already been done in the current version and it is now being validated by possible users. Also, an interactive graphic interface has been built in order to fulfill the requirements of an interactive simulation, as the SimOrg simulation is. The main graphic elements of the simulator are the following: the different data visions of the fictitious organization and the actions that the users could use to interact with the simulations.

Figure 7 illustrates a screenshot of the initial screen of the simulation system. On the top left-hand part of the figure, shown by arrow 1, it is possible to have a vision of the organization, described in a tree-structure. It shows the structure of all groups hierarchically. In Figure 7, for instance, the simulated organization has three groups, decision makers, designers and developers.

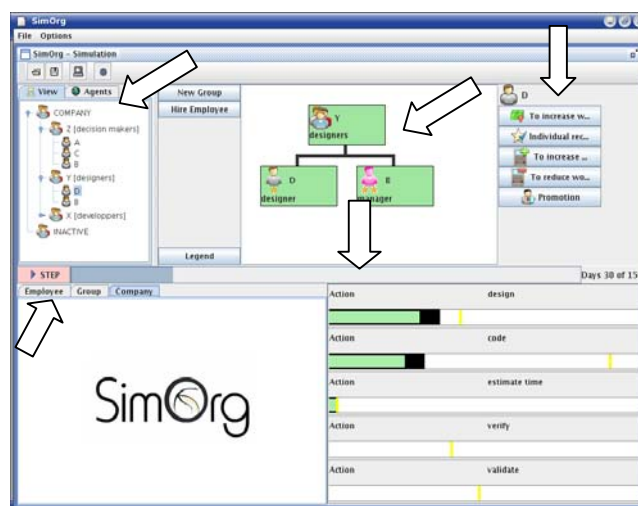


Figure 7 – Screenshot of the initial screen of the simulation.

On the top center part of the figure, shown by arrow 2, it is possible to see the organization chart, which describes the distribution of the agents into the groups. Also, it describes information about the state of the agents, such as the role of the agents in the group. The organizational chart can represent the whole company or parts of it. In Figure 7, for instance, it is shown the organizational chart of the designer group, which is composed of one designer (D) and a manager (E). In the organization chart and in the tree-structure, a user has the permission to change the structure of the organization, changing some agents to different groups and roles. Also, a user can perform user actions for the selected agent, pressing the left bottom of the mouse.

On the top right-hand part of the figure, shown by arrow 3, all the possible user actions are shown and, at any time during the simulation process, a user can choose to apply a user action. As already mentioned, the user actions can be done in order to improve the motivation level of an agent, aiming to increase the productivity of the organization.

On the bottom left-hand side, shown by arrow 4, there is a panel which contains information about agents, groups and the simulated organization. In this part, information about the agent history, such as: the roles that the agent performs within the group and activities of the groups, activities already performed by the agent, are described.

On the bottom right-hand side of Figure 2, shown by arrow 5, action graphics are described. These graphics represent the progress of all active actions of the system in relation to time. The main aim of these graphics is to show how early or late the agents are in the execution of the actions. The graphics show the estimated time of an action (yellow line) as well as the amount of time already used by the agents. In order to let the understanding of the graphic more intuitive, there is a variation of colors to represent the progress of the actions. The green color means that there is no delay of the action until the current iteration. On the other hand, the red color represents a delay in the progress of the action. Also, the black part of an activity means the optimal execution time. In using the real and optimal execution time, it is possible to analyze how late an activity is.

In addition to that, as an activity can be performed by a group of agents, in analyzing the agent history (bottom left-hand side), it is possible to detect which agent is having the most negative contribution to the delay in the execution of the activity. Also, for each activity performed by the agents, it is possible to visualize a graphic in which the contribution of each agent in the execution of the activity is described. It is shown when pressing the left button of the mouse in the activity required.

6. CONCLUSION

This paper presented a flexible framework for representing agents in a multi-agent system taking into account personality attributes for deriving the agent behavior. The flexibility refers to the ability to represent human personality into agents by the means of different models. This framework is based on the concept of working environment and reflects the static and dynamic structures and processes usually found in working organizations. As proof of concept, an existing simulation application, SimOrg, was adapted to the framework, having now the ability to represent personality in several ways.

The gain of the flexibility provided by the framework in this specific application is the possibility to easily adapt SimOrg to different companies. Indeed, as there is no consensus among psychologists about human personality, the Human Resource Department of company might work in a differently from another one. The simulation training tool SimOrg must then reflect the concept used by the company.

The ability to quickly change behaviors and personality traits has also been important tool for psychologists, since they can now analyze some important behavioral aspects of groups by prototyping by themselves their own experiments. These analyses will help the psychologists to define which aspects will have influence, for instance, in the productivity of a group and, then, to define which solutions could be done in order to improve the productivity of a working organization.

7. ACKNOWLEDGMENTS

This work has the financial support of CNPq (Brazilian Research Council), under process number 552431/2002-8.

8. REFERENCES

- [1] Church, A., and Waclawski, J. The relationship between individual personality orientation and executive leadership behaviour. *Journal of Occupational and Organizational Psychology*. 71(2):99-126. June 1998.
- [2] Cooper, M., and Sheldon, M. Seventy years of research on personality and close relationships: Substantive and methodological trends over time. *Journal of Personality*. 70(6):783, December 2002.
- [3] Digman, J. Personality Structure: Emergence of the Five-Factor Model. *Annual Review of Psychology*, 41:417-440, 1990.
- [4] Funder, D. *The Personality Puzzle*. Norton Ed. New York, US. 1997.
- [5] Georgeroff, M., Pell, B., Pollack, M., Tambe, M., and Wooldridge M. The Belief-Desire-Intention model of agency", *Proceedings of the 5th International Workshop on Intelligent Agent Theories, Architectures and Languages*, 1998.
- [6] Gilbert, N., and Troitzsch, K. *Simulation for the Social Scientist*. Open University Press. 1999.
- [7] Goldenstein, S., Menelaos K., Dimitris M., Guibas L., Aaron, E., and Goswami, A. Scalable Nonlinear Dynamical Systems for Agent Steering and Crowd Simulation. *Computers and Graphics*, 25(6):983-998. December 2001.
- [8] Hayes-Roth, B., and Rousseau D. A social-psychological model for synthetic actors. *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, Minneapolis, US, pp.165-172, 1998.
- [9] Jung, C. *Tipos Psicológicos*. (translation Orth, Lúcia) Ed. Vozes, Petrópolis, Brazil, 1991.
- [10] Loyall, B., Reilly, S., Bates, J., Weyhrauch, P. System for authoring highly interactive, personality-rich interactive characters. *Proceedings of the SIGGRAPH/EG Symposium on Computer Animation*, pp.59-68. 2004.
- [11] Millon, T. Reflections on psychosynergy: a model for integrating science, theory, classification, assessment, and therapy. *Journal of Personality Assessment*, 72(3):437-456, 1999.
- [12] Prada, R., and Paiva, A. Believable groups of synthetic characters. *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*. Utrecht, NL, pp.37-43, July 2005.
- [13] Revelle, W. Personality Processes. *Annual Review of Psychology* (also available at: <http://pmc.psych.nwu.edu/revelle/publications/AR.html>), 1995.
- [14] Salgado, J. Predicting job performance using FFM and non-FFM personality measures. *Journal of Occupational and Organizational Psychology*. 76(3):323-347. September 2003.
- [15] Ferber, J., and Gutknecht, O. A meta-model for the analysis and design of organizations in multi-agent systems. *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS98)*, pp. 128-135, Paris, France, 1998.
- [16] Gutknecht, O., and Ferber, J. The MADKIT Agent Platform Architecture. *Proceedings of the 1st Workshop on Infrastructure for Scalable Multi-Agent Systems*, pp. 48-55, June 2000.
- [17] Campos, A. and Hill, D., An Agent-Based Framework for Visual-Interactive Ecosystem Simulations. *Transaction of SCS*, 15(4):139-152, December, 1998.
- [18] Highsmith, J., *Agile Software Development Ecosystem*, The Agile Software Development Series, Addison Wesley Ed., 2002.