RPG–Profiler: a MAS for Role Playing Games Based Tests in Employee Assessment

Stefania Bandini, Sara Manzoni, Giuseppe Vizzari
Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano–Bicocca
Via Bicocca degli Arcimboldi, 8, 20126 Milano, Italy
{bandini, manzoni, vizzari}@disco.unimib.it

Abstract— This paper describes our experience in the design and implementation of a web–based computer system to support employee selection or assessment. Instead of traditional questionnaires the test is a *Role Playing Game* session, to reduce repetitiveness and predictability, and therefore to increase its effectiveness. The story is dynamically built by a multi–agent system, that chooses between available story modules according to the candidate's behaviour and on the kind of job he is being tested for. The psychological model is based on the jungian approach and the theory of archetypes. For sake of configurability, information about psychological model, scenographic modules and operative data is stored in a DBMS.

Index Terms—multi–agent systems, web–based systems, Role Playing Games, profiling

I. Introduction

The goal of a job interview is to discover or verify the competences of the candidate and to realize a partial psychological profile, in order to make meaningful predictions on his behaviour on the workplace. The former is an expensive task, requiring a competent interviewer, but the latter is not even objectively verifiable, and actually depends on the experience and the capability of the interviewer. Nevertheless problems that can arise from hiring a person that is not suited to the job or the work–environment can be serious. Many big companies arrange several job interviews for every candidate, to realize a sort of validation process for intermediate results, but it can be really expensive and time–consuming. Small firms often cannot afford it and eventually skip the psychological component of job interviews.

There are a number of different theories about psychological tests, ranging from the analysis of body posture to questionnaires. The latter are generally made up of a number of multiple–choice questions, with each answer pointing directly to one of the final categories in which the candidate has to be classified. The number of choices is generally fixed and sometimes the answer position is significative. For instance a majority of answers on the first position indicates that the candidate belongs to a particular category. This kind of test is generally quite repetitive and the information linked to every choice can be predictable. The candidate can understand the structure behind the questions and act consequently, reducing the test effectiveness.

The proposed approach uses *Role Playing Games* to test the candidate in a dynamic ludic environment, reducing repetitiveness and predictability, in an attempt to remove part of the barriers to a genuine communication. More than just pointing out a different method for test submission, we propose a psychological model for user profiling, based on *jungian archetipal theories*.

In this paper we will describe our experience in the design and implementation of a prototype for a Multi–Agent System (MAS) to support remote psychological profiling, exploiting this kind of approach. To allow remote interaction with the candidate, and for sake of configurability, the MAS is integrated into a distributed system, including a web–based user–interface manager and a database management system. The latter stores information related to the candidate and his profile but even the psychological model and the elements to dynamically compose the test. This information can thus be easily modified by competent personnel, through a suitable user–interface, allowing complete system configuration.

A MAS approach was considered for several reasons: first of all the problem is inherently distributed, with a profiling engine that processes data coming from a remote user according to information stored in a local database. The solution requires the use of data sources that are thus distributed and highly heterogeneous (i.e. a human user and a database management system). The dynamic composition of the test requires adaptivity and involves explicit reasoning about user's behaviour and test effectiveness. Moreover an agent based approach to system analysis and design enhanced its modularity, and even if it should not considered a sufficient condition to adopt agent–based technology [7], this should be taken into account.

The paper breaks down as follows: in the next section Role Playing Games and their non–ludic applications will be outlined, then the psychological model for user–profiling will be described and Section IV delineates how the test is dynamically built; following that, a brief description of the realized prototype will be given, with particular attention to the profiling engine; conclusions and future developments end the paper.

II. Role Playing Games and job interviews

It is not an easy task to explain what a game is, for it can involve so many different physical, mental, or social activities. Moreover one might call a game something that someone else could consider boring. An interesting attempt to define this term, with a specific reference to computer games design, can be found in [2].

Therefore we will not try to define Role Playing Games, and we will just focus on what we consider the main characteristic of this kind of ludic entertainment: the player "lives" those situations experienced by his character, his *alter ego* in a fantasy world. The player controls this character, and through it he lives adventures, situations, conflicts, just like a viewer of a good film but, having a certain degree of control over the events, in a much more intense way.

This kind of game and its implications on players' behaviour have been analyzed by psychologists (see, e.g., [3]), and there are

Archetype	**Goals**	**Fears**	**Gifts/Virtues**	**Shadows**
Innocent	Retain the *status quo*	Abandonment	Trust, optimism	Excessive trust, denial
Orphan	Regain safety	Exploitation	Realism, empathy	Cynicism, self–pity
Warrior	Win	Weakness	Courage, discipline	Violence, workaholism
Caregiver	Help others	Selfishness	Compassion, generosity	Martyrization, hidden purposes
Seeker	Search for a better life	Conformity	Autonomy, ambition	Perfectionism
Lover	Bliss	Loss of love	Passion, commitment	Jealousy, envy, dependence
Destroyer	Metamorphosis	Annihilation	Humility	Self-destructive behaviour
Creator	Identity	Inauthenticity	Individuality, vocation	Recklessness, obsessiveness
Ruler	Order	Chaos	Responsibility, control	Severity, oppressiveness
Magician	Transformation	Stagnation	Talent, personal power	Selfishness, irresponsibility
Sage	Truth	Deception	Wisdom, nonattachment	Presumptuousness, intolerance
Fool	Enjoyment	Nonaliveness	Joy, freedom	Self-indulgence, lazyness

TABLE I

A DESCRIPTION OF THE TWELVE ARCHETYPES WE CONSIDERED, INCLUDING THEIR GOALS, FEARS, GITFS AND POTENTIAL FLAWS AND ERRATIC BEHAVIOURS (*shadows*).

significative relationships between Role Playing Games and a kind psychotherapy called psychodrama [6]. In this kind of therapy the patient lives a controversial or psychologically problematic situation from points of view different from his own, to better understand them and live a cathartic experience.

The potential of this kind of game as an instrument in adult education and training has been recently analyzed (e.g. in [1]). For instance, a vendor could play the part of a client and interact with a trainer impersonating a vendor in a simulation of a real dialogue. In this way the former could realize how to modify his own approach by experiencing such a dialogue in a different role.

This kind of games are thus not actually limited to the ludic context. The proposed approach uses a Role Playing Game session as a metaphor to obtain information on the player's psychological profile basing on his in–game behaviour. The candidate is virtually inserted in a ludic environment and instead of answering to a set of "unlinked" questions he makes choices in the scenes of an interactive, dynamic story. In this kind of situation the candidate surely knows that he is being tested, but the distance between the job context and the game session can remove part of the barriers to a genuine communication and evaluation.

III. USER PROFILING

The desired profile has to be useful for job interviews sake, so we do not need, nor desire, a moral judgement of the candidate's behaviour in game or a medical diagnosis. Clearly the model is not an oracle, it just assumes that is possible to make meaningful predictions about the candidate's behaviour basing on his choices in a ludic environment. Anyway traditional tests are based upon a very similar assumptions, claiming to make predictions according to candidate's answers to a questionnaire.

To design a psychological model for this system we considered the elaboration of jungian theory of archetypes by Carol S. Pearson [9]. An archetype is a form (*Gestalt*), a sort of unconscious map of reference categories to help us understanding situations and behaving consequently. A list of the archetypes included in the model, with a brief description of their goals, fears, gitfs and potential flaws and erratic behaviours (*shadows*) is shown in Table I.

There are various relevant archetypes from the job interviewer's point of view: for instance a strong presence the archetype *Warrior* in a candidate gives him a good resistance to high–stress situations, and the absence of the *Caregiver* can be an issue for team–working.

A problem that might hinder the realization of a correct profile for the candidate is the use of direct links between answers and final categories. These links can be understood by the candidate, that can act consequently, not being himself. Moreover this kind of link is hard to define, and thus often results imprecise.

To avoid this problem an intermediate level of descriptors, less abstract than the archetypes (e.g. *courage, aggressiveness*) has been introduced to describe the candidate's profile. Every answer modifies the score of one or more descriptors, causing indirectly a modification of the final results, the presence of the archetypes in the candidate's profile. A similar profile must be defined for every archetype, in order to compute the distance between them and the one related to the candidate. This value gives an indication of the difference between the archetypes and the candidate, representing thus information about their presence in his profile. The relationship between answers and results of the test is not direct, less obvious, and thus the test is less predictable. Another positive side–effect is that without a direct link between choices and final categories there is no need to have a fixed number of choices, making the test less repetitive and further less predictable.

Descriptors have a different significance, positive or negative, for different job categories. For instance *charisma* is more important than *curiosity* for *leaders*; while both are positive traits,

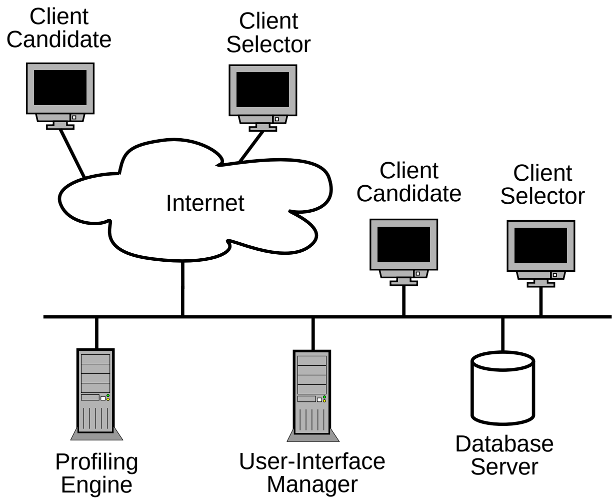


Fig. 1. The architecture of the prototype. Different modules (User–Interface Manager, Profiling Engine and Database Server) are placed in three different machines on a network, but it is not mandatory.

fear of confrontation **is quite negative for this kind of job position. This means that an indication of the kind of job must be given for every candidate and that the distance between different profiles must consider the relevance of each descriptor for the given job category. Moreover such distance should not be an absolute value, but a relative one, in order to indicate which profile is the best for a particular job category.**

The distance between two profiles $p1$ and $p2$, related to a particular job category, expressed by c, is thus defined as follows:

$$Dist(p_1, p_2, c) = \sum_{d \in D} \left(\Big(D_{val}(d, p_1) - D_{val}(d, p_2) \Big) \times r \right)$$

with

$$r = D_{relev}(d, c)$$

where D is the set of descriptors, D_{val} denotes the value of a descriptor for a given profile and D_{relev} maps the relevance of a descriptor for a job category. This function can be used to have an indication on the presence of an archetype in a candidate, but even to compare two candidates and know which one is more suited to a particular job category.

IV. Test building

This kind of considerations on the importance of descriptors are particularly relevant for the evaluation of the user's profile, but even for the story building. The test must focus on those descriptors that have a greater impact on the profile evaluation, and to do so each story must be built dynamically. The basic components of the test are story modules, small but coherent narrative units. Each module is a set of scenes (i.e. a single situation in which the player must act) and possible actions for the player. The internal representation of a module is a tree structure with scenes as nodes and available actions as connections between them. Scenes are thus linked and the player, choosing an action in every scene, descends into this tree and advances into the story. Every choice gives an indication on his profile, modifying the value of one or more descriptors.

The duration of the test can be expressed in terms of a minimum number of scenes that must be proposed to the candidate. A single story module is thus generally not enough for a complete test. In this case, when the candidate makes his choice in an ending scene of a module (a leaf of the related tree) another module must be chosen. For sake of story coherence every choice in an ending scene of a story module has an indication of what kind of module can be preferably inserted next, and modules are stored with a brief description. The module relevance is thus corrected adding a configurable fixed value to the modules whose description matches the preferred following module for the ending choice of the current one. This corrective is useful to avoid inconsistencies in the story, such as a character fleeing from a dangerous situation in a certain place and reappearing exactly there in a quiet scene.

Summarizing this process, at the beginning of the test, or after a story module has ended and the test is still not complete, the system chooses the module that is more relevant. A non–deterministic choice is done when more modules have the same relevance to the test. The relevance of a module m, related to a particular job category c, is obtained as follows:

$$M_{val}(m, c) = \left(\sum_{d \in D} Test(d, m) \times D_{relev}\Big(d, c\Big) \right) + k$$

with

$$k = \begin{cases} \mathbf{i} & Mtype(m) = \textbf{Preferred following type} \\ 0 & \textbf{otherwise} \end{cases}$$

where $Test(d, m)$ assumes value 1 if the descriptor d is tested by the module m and 0 otherwise, $Mtype(m)$ denotes the type of the module m, and i is a configurable constant value.

V. Prototype overview

The main features of the desired system are the possibility to make remote profiling via a web–based interface, and its complete configurability. According to these functional requirements the architecture of the realized prototype provides a dedicated module for user interface handling and another for the storage of the configurable information. User profiling is a completely different activity, requiring knowledge representation and inference capabilities, so this task has been assigned to a dedicated module. The global architecture of the prototype is shown in Figure 1: system modules (User–Interface Manager, Profiling Engine and Database Server) are located in three different machines on a network, but it is not mandatory.

The User–Interface (UI) Manager is a set of *Servlets* and *Java Server Pages* (JSP), that produces HTML pages to show a textual description of the scene, an image, a selection for the possible behaviours and a button to confirm the user's choice. A screenshot of the candidate user–interface is shown in Figure 4. This module also handle selector user interface, to grant access to all the configurable information, for instance story modules, psychological model and choices' evaluations.

Information needed to compose the pages for candidate user interface is obtained from the Profiling Engine, that is thus responsible for story building, but even for user profiling. This module is a multi–agent system based on the Zeus toolkit and will be described in the following section.

The last component of the system is responsible for the storage of the relevant information needed by previous modules. As shown in Figure 2 stored data concerns the psychological model (classes descriptor, archetype, archDesVal, categ and descrRelev), story modules (classes scenogModule, moduleType, moduleDetail, choice and evaluation) and operative information (classes person, descValue, role and persCateg). The Object Oriented Data Base Management System (OODBMS) Versant [11] was chosen, mainly because of its seamless integration with Java applications.

Candidates and selectors access the system via a web–browser, that communicates with the User–Interface Manager through Hyper Text Transfer Protocol. The Profiling Engine provides a Remote Method Interface (RMI) server, used by the UI handler to

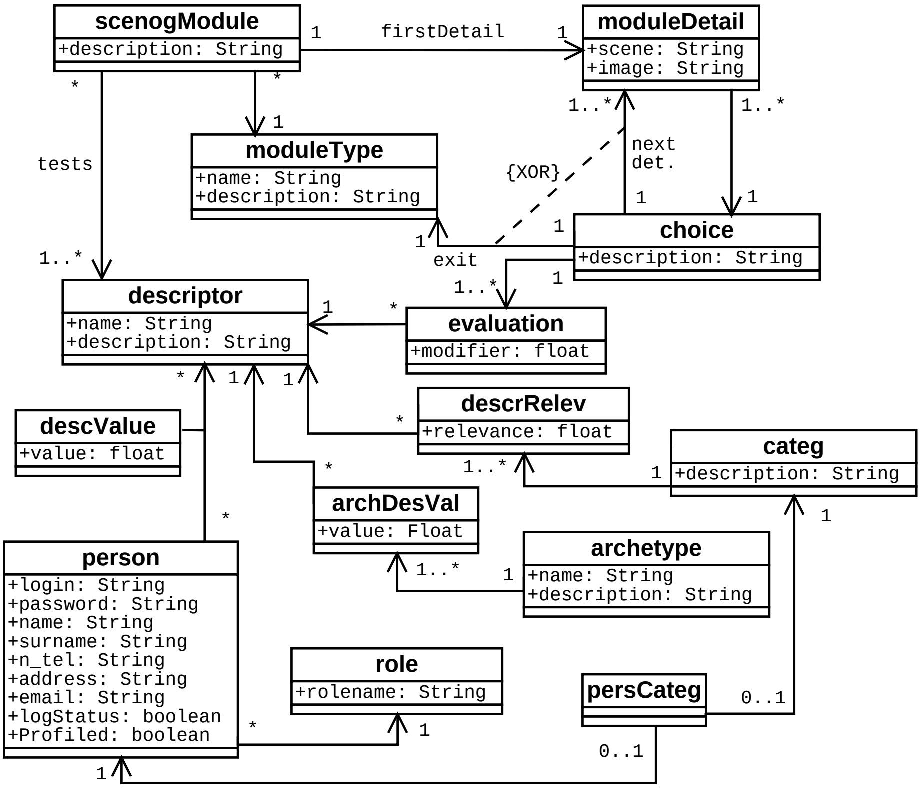


Fig. 2. Class diagram of the database. Stored information is related to the psychological model, to story modules and operative data.

obtain information to compose the web pages. The Profiling Engine processes data obtained by the Versant OODBMS via a Java API, provided by the database in the form of a class hierarchy.

VI. Profiling Engine

The Profiling Engine is responsible for

- story flow management and test composition;
- evaluation of user choices;
- profile management.

These are very different functions and, even if they are related to a single role in the global system, it makes sense to assign them to different sub–modules. To fulfill these tasks the Profiling Engine has to communicate with the UI Manager and the Database Server, that could hosted by different physical machines, so the information sources and users are to be considered as distributed over a network. Moreover they are obviously highly heterogeneous, being a human and a DBMS. The test composition task and strategy, as seen in Section IV, requires adaptivity and involves explicit reasoning about user's behaviour, job category and test effectiveness. A multi–agent approach thus seems rather appropriate to model this system.

Instead of starting system design and development from scratch several toolkits and frameworks facilitating analysis, design and implementation of multi–agent systems were considered (according to criteria similar to the ones used in [10]) and Zeus [8] was chosen, mostly because of the amount and quality of available documentation. It supports the design and development of systems made up of collaborative, deliberative agents, and provides a complete environment, based on the Java platform, supporting agent adressing, communication, reasoning and cooperation.

The internal architecture of the MAS implementing the Profiling Engine, displaying collaborations between agents, is shown in Figure 3. The system actually needs a support agent that is not shown in the figure because it is not involved in user profiling: the Agent Name Server. This is a simple reactive agent, responsible for the translation between agent names and IP addresses. In fact, even if in Figure 1 the Profiling Engine is shown as a single server, agents can be located in different machines over a network and communicate via TCP/IP protocol.

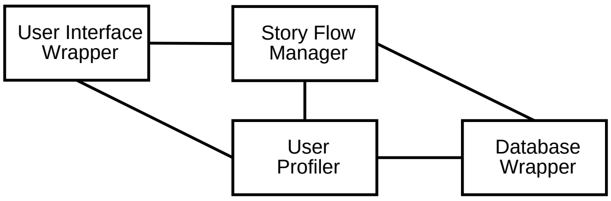
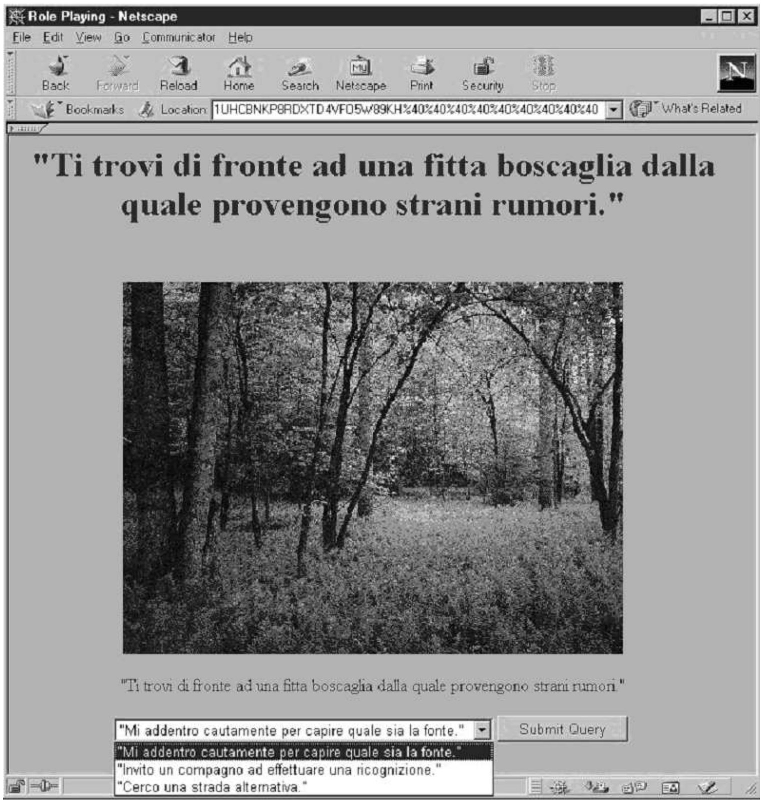


Fig. 3. The internal architecture of the Profiling Engine, showing collaborations between agents.

Putting together legacy systems (i.e. non agent software) with a MAS can imply serious integration issues, but it is a subject of crucial importance, considering for instance the installed base of database management systems or the pervasiveness of the web. Moreover agent technology has often been considered an instrument to promote software interoperation (see, e.g., [5]). In this case both the data storage facility and the front–end of the system required the development of a specific wrapping agent, respectively to translate persistent data to a representation compatible with the MAS internal ontology, and to a suitable format for the UI handler. On the other hand a MAS approach supplied proper abstractions for system analysis and modeling (see, e.g., [12]), and the chosen framework provided a valuable environment supporting agent development, deployment and operation.

A. User Interface Wrapper

This agent wraps around the user interface modules, and acts as an interface between servlets/JSPs and the agent community.


Role Playing - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: 1UHCBNKP8RDXTD4VFD5W89KH%40%40%40%40%40%40%40%40%40
What's Related
Submit Query

"Ti trovi di fronte ad una fitta boscaglia dalla quale provengono strani rumori."

"Ti trovi di fronte ad una fitta boscaglia dalla quale provengono strani rumori."

"Mi addentro cautamente per capire quale sia la fonte."

"Mi addentro cautamente per capire quale sia la fonte."
"Invito un compagno ad effettuare una ricognizione."
"Cerco una strada alternativa."

Fig. 4. A screenshot of the candidate user–interface, with a textual description of the scene, an image, a selection for the possible behaviours and a button to confirm the user's choice.

These different entities can be placed on different computers on a network, so the UI wrapper implements a RMI server, to grant UI modules a remote access to the agent community.

At the beginning of a test sequence it receives a login message from a servlet that verifies the user account, and informs the User Profiler agent to prepare for his evaluation. After that it requires the first scene of the test to the Story Flow Manager. Every time the user makes a choice it sends it to the story manager and requires how to proceed.

B. Database Wrapper

This agent is a wrapper to the Database Server, and acts as an interface between the agent community and the database server, translating requests into a queries and results into a representation compatible with the MAS ontology. To do so it uses some classes from the API provided by the Versant OODBMS suite, that grant remote access to the database.

When activated it provides the Story Flow Manager and the Profiling Manager with information about the psychological model that will be used in their activity. After that when it receives requests from agents, it queries the database and sends the retrieved data back to the requiring agent. It is also responsible of storing the user's profile, sent by the Profiling Manager.

C. User Profiler

This agent is responsible for choices evaluation and candidate profiling. When it receives a login notification it requires user's data to the Database Wrapper, prepares a clean (i.e. neutral) user profile, then notifies the Story Flow Manager about the preferences for the choice of story modules. During the test it receives information about user's choices from the story manager, requires relative evaluation elements to the database handler and applies them to the user profile. It also controls the number of scenes proposed to the candidate and when it reaches a configurable value it stops the test, sending a message to the User Interface Wrapper and sending the final profile to the Database Wrapper for storage sake.

D. Story Flow Manager

This agent is responsible for story flow, and thus even story building. It receives information about choices and scene requests from the User Interface Wrapper and provides it with scenes directly pointed by candidate's choice or with the first scene of a new story module, chosen according to the strategy explained in Section IV. Information about user's choices and a login notification are dispatched to the User Profiler, for evaluation purpose. Information about scenes, available choices, and story modules are obtained by the Database Wrapper.

While other agents just act in reaction to requests received from the external environment or other entities, this one has a goal: to realize a coherent story and, at the same time, put to the test the most relevant descriptors for the candidate's job category. To achieve this goal it is equipped with a production system that realizes its own strategy, and an inferential engine based on the RETE algorithm [4].

VII. Conclusions and future developments

A prototype of the system was implemented and tested in a heterogeneus, distributed environment. Story modules were written to test the system but are not suited for real psychological evaluation. The prototype was useful to demonstrate the potential benefits of this kind of approach and actually started a project involving a company from the human–resources management area. A field–test of the system, and a comparison with traditional assessment methods, would be very useful to have an evaluation of the psychological model and the test submission procedure (i.e. a Role Playing Game session).

However this kind of system could be useful even in different areas: for instance this game could be proposed by e–commerce websites to realize commercial profiles of their customers. This application clearly requires a different kind of analysis on the psychological model, focusing on how various archetypes relate to customer profiles and habits.

Further work on the psychological model could also focus on various kinds of interaction between the different archetypes. This could be useful to define the guidelines for a small–scale social dynamics simulator, a tool that could use the profiles realized with this system as input data describing members of a group. A possible application of this kind of instrument could be in the human–resource area, and more specifically for team–building purpose.

References

[1] S. Capranico (1997), "Role–Playing. Manuale ad uso di formatori ed insegnanti", Raffaello Cortina, Milano.

[2] C. Crawford, (1982), "The Art of Computer Game Design," Washington State University.", REV.7.17.97 (Electronic version from 1997). http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html.

[3] N. A. Douse and I. C. McManus (1993), "The Personality of Fantasy Game Players", *British Journal of Psychology*, Vol. 84 (4), pp 505–509.

[4] C.L. Forgy (1982), "Rete: a fast algorithm for the many pattern/many object pattern match problem", *Artificial Intelligence*, Vol. 19, pp 17–37.

[5] M. R. Genesereth and S. P. Ketchpel (1994), "Software agents", *Communications of the ACM*, Vol. 37 (7), pp 48-53.

[6] J.L. Moreno (1975), "The theatre of spontaneity", Vol. 1, Beacon Press, New York.

[7] H.S. Nwana and D.T. Ndumu (1999), "A Perspective on Software Agents Research", *The Knowledge Engineering Review*, Vol. 14 (2), pp 1–18.

[8] H. S. Nwana, D. T. Ndumu, L. C. Lee, J. C. Collis (1999), "ZEUS: A Tool–Kit for Building Distributed Multi–Agent Systems", *Applied Artifical Intelligence Journal*, Vol. 13 (1), pp 129–186.

[9] C. S. Pearson (1991), "Awakening the hero within: twelve archetypes to help us find ourselves and transform our world", Harper SanFrancisco, New York.

[10] P. M. Ricordel and Y. Demazeau (2000), "From analysis to deployment: a multiagent platform survey", *Proceedings of 1st International Workshop on Enginnering Societis in the Agents World (ESAW), ECAI'2000*, pp 93–105.

[11] Versant Corp. (2001), Versant Home Page, http://www.versant.com.

[12] M. Wooldridge, N. J. Jennings, D. Kinny (2000),"The Gaia Methodology for Agent-Oriented Analysis and Design", *Journal of Autonomous Agents and Multi-Agent Systems*, Vol. 3(3), pp 285–312.