Towards Adaptive e-Learning / e-Teaching on the Web

Víctor López-Jaquero, Antonio Fernández-Caballero, Francisco Montero, José Pascual Molina and Pascual González

Laboratory on User Interaction & Software Engineering (LoUISE)
University of Castilla-La Mancha, 02071 Albacete, Spain
{victor|caballer|fmontero|jpmolina|pgonzalez}@info-ab.uclm.es

Abstract. With the first stages of e-learning technologies maturity, this kind of technologies have spread to improve learning experience with a number of well-known key benefits. At the same time, some drawbacks have been found. One of them is the lack of integration of the diversity in the skills and preferences of the students. In this paper a proposal is introduced to capture users' preferences and skills by means of a multi-agent system to reflect the high diversity in the student mass in the way they are taught. The proposal is based on usability metrics and social computing as a mean to acquire the student model needed in order to adapt navigation, contents and presentation.

1.   Introduction

New information technologies offer new tools that can fully modify the way we nowadays think of teaching. Multimedia presentation and interactivity reinforce understanding and improve motivation. On the other hand, the possibility of building learning communities fosters human interaction, knowledge sharing, and social capabilities. Nowadays, e-learning is becoming more popular because of the low cost of the equipments and the beginning of the maturity stage of these kinds of e-learning technologies. However, still much work is left to be able to adapt the e-learning system to students in a seamlessly way.

Designing adaptive e-learning systems implies that the design of the interaction between the system and the systems should be personalised. However, human-computer interaction (HCI) in traditional application development has been focused on the interaction between a task and a single user interface designed for a single kind of user, making no distinction between the diverse skills or preferences they might have (figure 1a). This approach has been used for some years in intelligent tutoring systems. However a logical evolution should lead interaction to a development model where the differences on the stereotypes of the students are taken into account. There are different kinds of users, and that is a fact we cannot ignore. Human society is full of diversity and that must be reflected in the way we teach (figure 1b). However, one step forward in interaction design is required in order to translate this diversity into application development. Adding support for different user profiles is, of course, more accurate than developing for a single kind of user. But the real thing is that students are all a little bit different. This student may match a student profile, but with his own particularities, leading to the concept of specialization (figure 1c). Thus, we need to engage users in a new kind of interaction concept where user interfaces are tailored-made for each user and where the user interfaces are intelligent and adaptive [18] [19]. These new concept of interaction should take a real step towards providing users with what they really need and prefer. On the other hand, having a single system capable of supporting a variety of learner types will facilitate ease of distribution of e-learning materials and provide a standardised framework to support system scalability with respect to ever-changing learning materials.
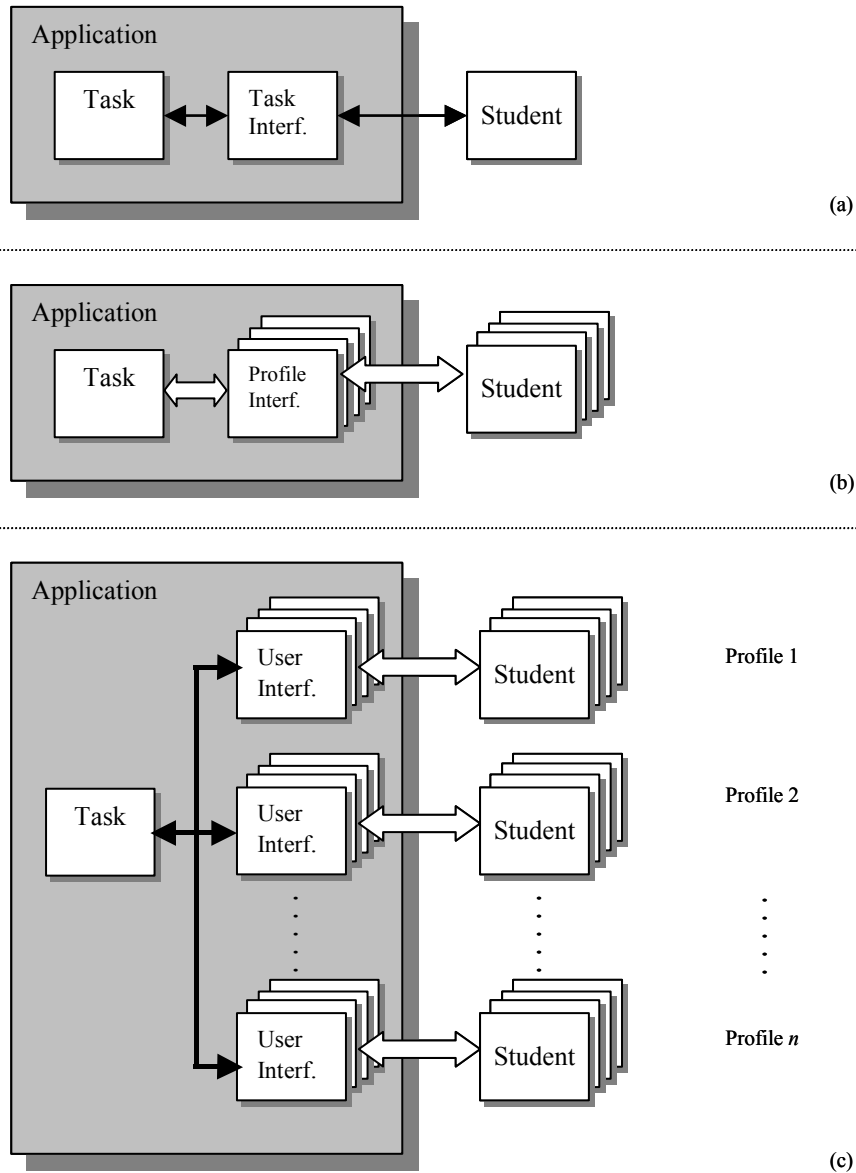
Fig. 1. (a) Unity, (b) diversity, and (c) specialization in computer interaction.

In this paper we propose to include two up-to-date separate concepts, namely, social computing and usability metrics, in intelligent interaction agents to enhance user-centred, adaptive e-learning / e-teaching applications on the Web. As already clearly stated [25], "people should not have to change radically to 'fit in the system'; the system should be designed to match their requirements".

Social computing has been largely studied over the last decade, directly or indirectly, as it may be appreciated by the great amount of papers on the topic (e.g. [12] [22] [23] [4]). Adaptivity is also a today's hit [2] [26] [27]. And, there have been some approaches to fit these ideas into concrete architectures [26] [5] [8].

2. Social Computing

Social computing refers to the application of sociological understanding to the design of interactive systems [7]. Traditional approaches – based perhaps on functional specifications or on laboratory-based usability studies – tend to be disconnected from the lived detail of the work. Accountability is an idea essential to social computing in the analysis and development of software systems. Accountability means that the interface is designed so as to present, as a part of its action, an "account" of what is happening. The goal of the account is to make the action of the system concrete as a part of an ongoing interaction between the system and the user. Applications should log the interaction history for each user. All these logged accounting information will be gathered and stored in a knowledge base. Thus, the user interface presented to the user could use those accounting data as an input for customizing the user interface presentation, in order to reflect the user preferences.

3. Usability Metrics

Usability metrics are software quality metrics with a long history of successful application in software engineering [1] [8] [11]. But, metrics also carry risks [3]. No simple number can completely represent anything as subtle and complex as the usability of a software system, but numbers can sometimes create the illusion of understanding. Metrics for usability can be thought of as falling into three broad categories: preference metrics, which quantify the subjective evaluations and preferences of users, performance metrics, which measure the actual use of working software, and predictive metrics, or design metrics, which assess the quality of designs and prototypes.

One of the most popular ways to assess usability is to use preference metrics [24]. User satisfaction is a component of usability and also an important factor in success in the marketplace. Preference metrics are one of the pillars for user interface customization. However, because of their intrinsic characteristics, they are difficult to assess at run time. Usually, questionnaires are used for evaluating these preference metrics. However, there are some preference metrics, such as the manipulation artefact used when commanding tasks (keyboard, menus, and toolbars) that can become especially useful for capturing user preferences.

On the other hand, performance metrics are indices of various aspects of how users perform during actual or simulated work. User performance is almost always measured by having a group of test users perform a predefined set of test tasks while collecting time and error data [21]. Typical quantifiable usability measurements include: the time users take to complete a task; the number of tasks of various kinds that can be completed within a given time limit; the ratio between successful interactions and errors; the time spent recovering from errors; the number of user errors; and so on [21]. Of course, only a subset of these measurements would be collected during any particular study. Performance metrics are especially useful for assessing overall usability. One important point for this kind of metrics is that most of them can be evaluated at run time in a simple manner. Performance metrics are one more input parameter to advance towards interfaces adapted to the user. Our proposal is to forget user interfaces where the user must adapt to a given and fixed interface.

A central goal in software metrics is the prediction of software characteristics based on other metrics of the software or its production process. In [29], Rosenberg addresses this problem by providing a simple methodology for the predictive evaluation of metrics, one which reflects what most statisticians would recommend in general terms. Predictive metrics are estimators or predictors of some one or more aspects of the actual performance that can be expected once a system has been implemented and put into use.

4. An Interaction Multi-Agent System

To build a system that enhances the user's motivation in Web based applications [6] [10], it would require nearly constant surveillance of the user, and at least strong artificial intelligence – knowledge, intelligence, memory, insight, and reasoning ability equivalent to that possessed by an intelligent being. A quite recent study [15] integrates that the person feeling frustrated suddenly has diminished abilities for attention [16], memory retention [14], learning [17], thinking creatively [13], and polite social interaction [9], among other things – as well as a penchant for getting more frustrated in the immediate future. Our proposal is to design and build interaction multi-agent systems that significantly help users in their relationship to the WWW. Humans are much more than information processors. Humans are affective beings, motivated to action by a complex system of emotions, drives, needs, and environmental conditioning [20].

The ultimate goal for HCI must be the creation of user interfaces based on each individual user preferences. Those preferences can be captured initially, to a certain extent, in analysis development stages. By using those captured data, user profiles can be created in accordance with the identified user stereotypes. However, user advances, and his preferences change. One rookie user will become an average user, and finally an expert user, as he gets more familiar with the system. Thus, the environment should reflect these changes in both user skills and preferences. Intelligent agents with the ability to capture those changes are required.

Figure 2 illustrates the user-centred interaction multi-agent architecture proposed for the adaptivity on the Web. The Multi-Agent System (MAS) monitors the interaction between the user and the user interface, and captures two different parameters. On one hand, the MAS logs the interaction history in an accounting process. On the other hand, the agent assesses preference metrics, such as the preferred interaction artefact for each action (keyboard, menu, toolbar, etc).
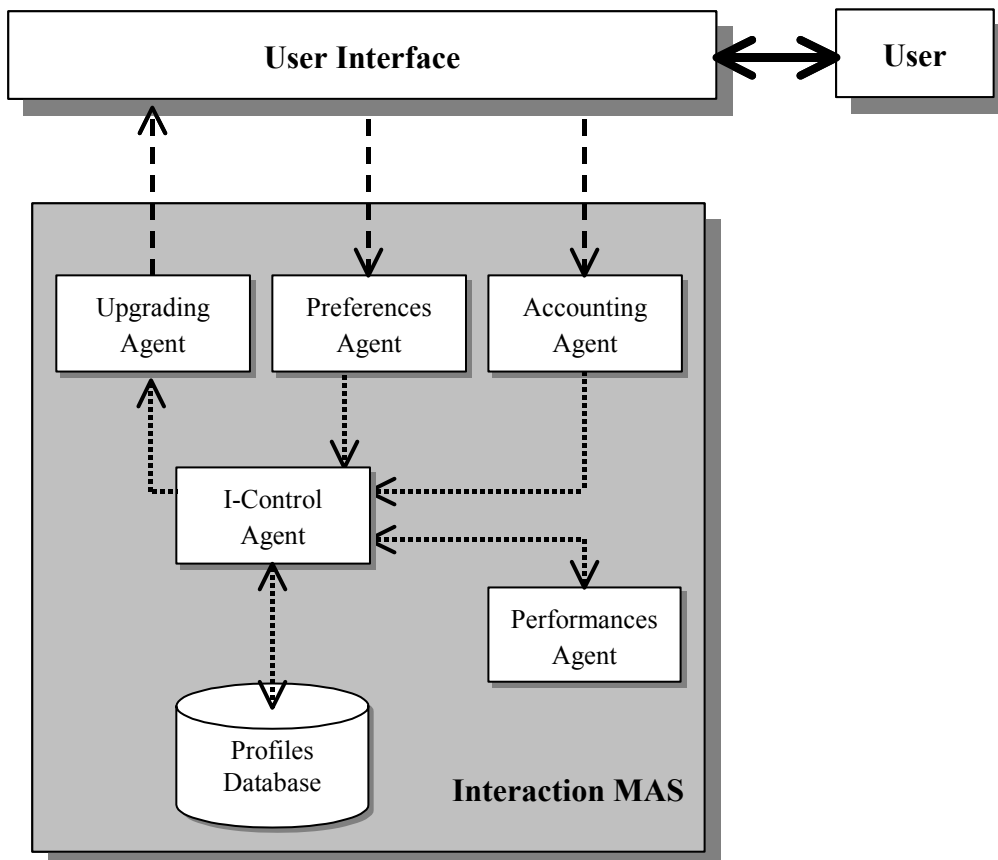

Fig. 2. The interaction MAS architecture.

Besides the described parameters, the user-centred adaptive interactive MAS assesses some behavioural systems issues (performance metrics), the error ratio regarding the number of correct interaction actions, or the time spent performing a task or recovering from an error. All these data are processed and stored in a knowledge base that the agent uses to decide the actions required in order to achieve its goals; to find out the possible changes applicable to the user interface to get it closer to what the user is expecting. Thus, a new user interface is created tailor-made in accordance with the skills, experience and preferences the user shows.

5. Adaptive E-Learning and E-Teaching on the Web

The architecture proposed so far is being tested in an on-line Web-based e-learning system as an intelligent tutoring system (ITS) for an Engineering course taught at the Polytechnic Superior School of Albacete, University of Castilla-La Mancha.

One of the main goals is that the alumni learn more and better, that is to say, to be able to structure learning matter in such a way to facilitate the learning facilities. One characteristic to take into account in learning is the rhythm the student is able to learn. Thus, a ITS has to adapt the rhythm it introduces the concepts to the learning rhythm of each student (for instance, to show more or less exercises, to show more or less tests, etc.). Another aspect widely considered in learning theory is reinforcement by rewarding a correct answer and punishing a failure. Rewarding and punishment can be carried out by means of messages, sounds, etc. Another goal in our environment is to enhance teaching as well as learning. One of the main problems a professor faces when teaching is that he does not know the skills of his students. Our proposal leads to conclusions that "teach how to teach".
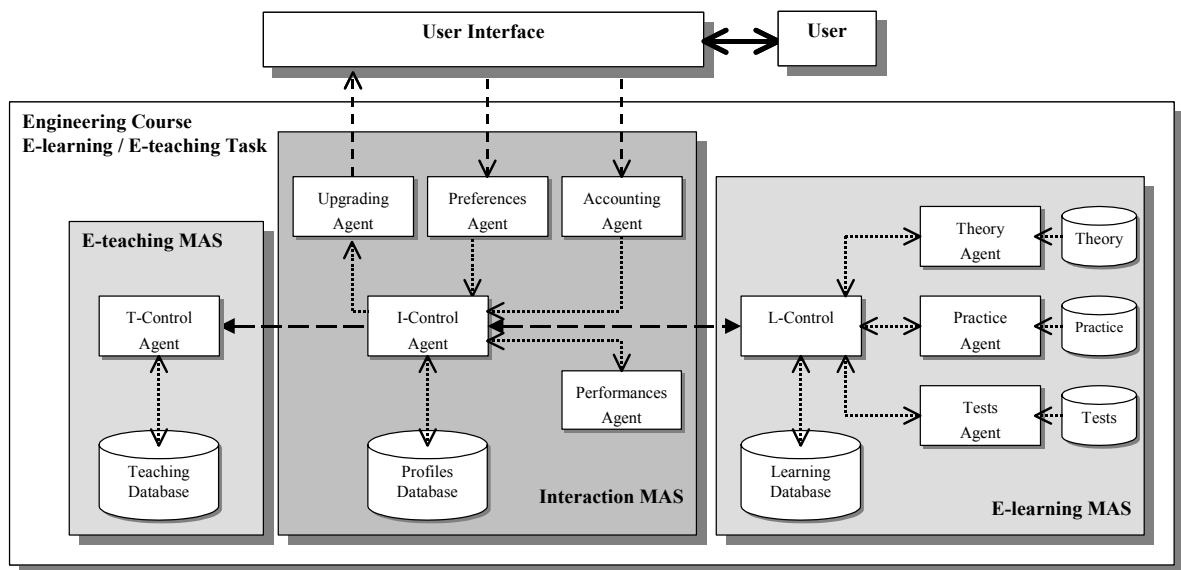

Fig. 3. The adaptive Engineering course task architecture.

In our learning system on the Web (figure 3) we have introduced three MAS: (1) The *Interaction MAS*, which captures the user preferences by means of some usability metrics (affect, efficiency, helpfulness, control and learnability). The contents shown to the user are created in accordance to the captured preferences and behaviours. (2) The *Learning MAS* composes the contents for the user in accordance to the information collected by the *Interaction MAS*. (3) The *Teaching MAS* is one of the most important contributions in our experience. It offers recommendations of how to enhance the layout of the Engineering course.

The user (the student) is in front of the user interface. From the interaction of both entities, modelled by the *Interaction MAS*, different metrics that are stored in a *Profiles Knowledge*

Database (KDB) are collected. This database contains the different profiles as a result of the use of the system by different students, with different aptitudes, motivations, etc.

The multi-agent system for the learning (*Learning MAS*), gets data obtained from the profiles (analysis of the distinct metrics captured) and adequates the contents shown to the concrete student that accesses the Web site. On the other hand, the multi-agent system for teaching (*Teaching MAS*) obtains measures that permit to get recommendations to enhance the course. Finally, the course has been decomposed into theory, exercises and tests.

5.1. Learning Multi-Agent System

The *Learning MAS* appears from the general goal to maximize the course learning. The *learning control agent* communicates bi-directionally (asks for and receives information) with the *theory agent*, the *exercises agent*, the *tests agent* and with the *interaction control agent* (*Interaction MAS*). This agent asks for/receives Theory Web Pages to/from the *theory agent*, asks for/receives Exercises Web Pages to/from the *exercises agent*, asks for/receives Tests Web Pages to/from the *tests agent* and communicates (through the *interaction control agent*) with the *performance agent* to record the performance of the student in order to decide if he needs a reinforcement. If the student needs some kind of reinforcement the *learning control agent* will elaborate a plan with the material that has to be shown to the student. In order to determine if the student needs reinforcement the *performance agent* will have access to a *KDB* where the minimum requisites for each subject are stored (quantity of exercises to be initially shown to the student, how many exercises the student has to answer correctly, and in how much time, maximum time to correctly answer an exercise, etc.).

The *theory agent* is constantly waiting for the *learning control agent* to ask for a Theory Web Page. When this occurs, it looks for the proper theory page and sends it to the *learning control agent*. The *exercises agent* is autonomous as it controls its proper actions in some degree. The agent by its own means (pro-active) selects the set of exercises to be proposed in the subject studied by the student and adds to each exercise the links to the theory pages that explain the concepts related to the exercise. It sends to the *learning control agent* a Web page containing the exercises proposed. The *tests agent* is continuously listening to the *learning control agent* until it is asked for Tests Web Pages. The agent by its own means (pro-active) goes on designing a set of tests for the subject the student is engaged in. These tests will be shown to the student in form of a Web.

5.2. Teaching Multi-Agent System

The *Teaching MAS* is the result of the second general goal fixed, namely, to maximize the teaching capacity of the course. The *Teaching MAS* will be collecting the goodness or badness of the parameters defined for the learning system. The *Teaching MAS* is pro-active in the sense that it will be providing recommendations to the teacher on those parameters.

5.3. Interaction Multi-Agent System

The *Interaction MAS* has been conceived to facilitate the adaptive communication between the system and the user. The *interaction control agent* tells the *upgrading agent* what the user preferences are, as obtained by the *preference agent*, and which has to be the next Web page to be shown (*learning control agent* of *Learning MAS*). When speaking about the preferences of the student, we mean the type of letter, the colour, the icons, etc., the user prefers. The information collected is stored in the *Profiles KDB*. All information concerning time-related parameters and some of the user's behaviours are obtained through the *performance agent* and the *accounting agent*.

The *preference agent* perceives the interaction of the user with the user interface and acts when the user changes his tastes. The *preference agent* is continually running to know the student's preferences at any time.

The *performance agent* calculates the performance metrics when the student leaves the system (at the end of a working session) y goes evaluating everything the student does in order to know if he needs reinforcement. It is autonomous and pro-active; as it may calculate metrics at the same time the student performs other tasks. Some of the metrics the *performance agent* handles are: for each Theory Web Page, the mean time alumni spend there; for each exercise Web page, the mean punctuation obtained by the alumni, as well as the time spent to get the correct answer; for each Tests Web Page, the mean time spent to answer all questions, and the mean punctuation obtained in the tests.

The *accounting agent* perceives the interaction between the student and the user interface and acts (gets information) when the student changes to another Web page, scrolls up and/or down, performs an exercise or a test, etc.

Finally, the *upgrading agent* is constantly waiting for the *interaction control agent* to ask to update the user interface with the new information to be shown to the student (to show another Web page or to show the same Web page but changed to the new tastes of the student).

6.   Conclusions

In this paper we have proposed an architecture that considers the high diversity of users' skills and preferences: a user-centred and adaptive interaction multi-agent system. It is being applied to the e-learning / e-teaching of an Engineering course. This architecture is inspired in social computing and usability metrics. The systems is able to support adaptation to the different students' skills and preferences to adapt contents, navigation and presentation. Furthermore, it is also able to make suggestions about possible errors in teaching strategy.

Acknowledgements

References

[1] Card, D., Glass, R., 1990. *Measuring Software Design Quality*. Prentice-Hall.

[2] Cardon, A., Lesage, F., 1998. Toward adaptive information systems: considering concern and intentionality. *Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'98* .

[3] Constantine.L.L., Lockwood L.A.D., 1999. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Addison-Wesley.

[4] d'Inverno, M., Luck, M., 2000. Sociological agents for effective social action. *Proceedings of the Fourth International Conference on Multi-Agent Systems*, IEEE Computer Society, 379-380.

[5] Eisenstein, J., Rich, C., 2002. Agents and GUIs from task models. *International Conference on Intelligent User Interfaces*. ACM Press, 47-54.

[6] Fernández-Caballero, A., López-Jaquero, V., Montero, F. , González, P., 2003. Adaptive Interaction Multi-agent Systems in E-learning/E-teaching on the Web. Web Engineering: International Conference, ICWE 2003, Oviedo, Spain, July 14-18, 2003. Proceedings. J.M. Cueva Lovelle, B.M. González Rodríguez, L. Joyanes Aguilar, J.E. Labra Gayo, M. del Puerto Paule Ruiz (Eds.). Springer Verlag, LNCS 2722, pp. 144-154.

[7] Garfinkel, H., 1967. *Studies in Ethnomethodology*. Polity Press.

[8] Gilb,T., 1977. *Software Metrics*. Winthrop Publishers, Inc., Cambridge MA.

[9] Goleman, D., 1995. *Emotional Intelligence*. Bantam Books.

[10] González, P., Montero, F., López-Jaquero, V., Fernández-Caballero, A., Montañés, J., Sánchez, T., 2001. A Virtual Learning Environment for Short Age Children. *Proceedings of the IEEE International Conference on Advanced Learning Technologies, ICALT2001,* IEEE Computer Society, 283-284.

[11] Henderson-Sellers, B., 1996. O-O Metrics: Measures of Complexity. Prentice-Hall.

[12] Horovitz, E., 1999. Uncertainty, action, and interaction: In pursuit of mixed-initiative computting. *Intelligent Systems*. Computer IEEE Society. September / October, 17-20.

[13] Isen, A.M., Daubman, K.A., Nowicki, G.P., 1987. Positive affect facilitates creative problem solving. *Journal of Personality and Social Psychology,* Vol. 52, No. 6, 1122-1131.

[14] Kahneman, D., 1973. *Attention and Effort*. Prentice Hall.

[15] Klein, J.T., 1999. Computer Response to User Frustration. MIT.

[16] Kitayama, S., Niedenthal, P.M., 1994. Heart's Eye: Emotional Influences in Perception and Attention. Academic Press.

[17] Lewis V.E., Williams, R.N., 1989. Mood-congruent vs. mood-state-dependent learning: Implications for a view of emotion. In Kruiken, D. (ed.) Mood and Memory: Theory, Research, and Applications (vol. 4) of Special Issue of the *Journal of Social Behavior and Personality*, Vol. 2, 157-171.

[18] López-Jaquero, V., Montero, F., Fernández-Caballero, A., Lozano, M.D., 2003. Towards adaptive user interfaces generation: One step closer to people. *Proceedings of ICEIS 2003*.

[19] López-Jaquero, V., Montero, F., Fernández-Caballero, A., Lozano, M.D., 2003. Usability metrics in adaptive agent-based tutoring systems. *Proceedings of HCI International 2003*.

[20] Myers, D.G., 1989. *Psychology*. Worth Publishers.

[21] Nielsen, J., 1993. *Usability Engineering*. Academic Press.

[22] Oliveira, E., 2001. Agent's advanced features for negotiation and coordination. *Multi-Agent Systems and Applications*. LNAI, Vol. 2086. Springer, 173-186.

[23] Petta, P. Trappl, R., 2001. Emotions and agents. *Multi-Agent Systems and Applications*. Lecture Notes in Artificial Intelligence, Vol. 2086. Springer, 301-316.

[24] Porteous, M., Kirakowski, J., Corbett, M., 1993. *SUMI User Handbook*. University College Cork, Ireland.

[25] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T., 1994. *Human-Computer Interaction*. Addison-Wesley.

[26] Sycara, K.P., 1998. Levels of adaptivity in systems of coordinating information systems. *Cooperative Information Agents II*. LNAI, Vol. 1435. Springer , 172-189.

[27] Suzuki, J., Tashiro G., Abe, Y., Yamamoti, Y., 1998. Persona: a framework to provide adaptive presentation for web documents. *Proceedings of the IPSJ Summer Programming Symposium, IPSJ Prosym '98*, 101-108.

[28] W3C, 2002. http://www.w3.org/WAI/

[29] Rosenberg, J., 1998. A methodology for Evaluating Predictive Metrics. *Proceedings of the 5th International Symposium on Software Metrics*. Maryland. USA, 181.