# Social Computing and Usability Metrics: Toward User-Centered and Adaptive Interaction Agents

Victor Lopez-Jaquero, Antonio Fernandez-Caballero, Francisco Montero, and Pascual Gonzalez

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

**Abstract.** 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 a user-centered, adaptive human-computer interaction (HCI). Social computing refers to the application of sociological understanding to the design of interactive systems. We introduce accountability as an idea essential to social computing in the analysis of software systems. Another idea underlying social computing is that, by configuring the space in different ways, different kinds of behavior can be supported. Usability metrics are software quality metrics with a long history of successful application in software engineering. We introduce preference metrics, which quantify the subjective evaluations and preferences of users, and performance metrics, which measure the actual use of working software, as other parameters that enable user interface adaptation. From all terms, a novel user-centered and adaptive interaction agents' architecture is proposed.

## 1 Introduction

Many things about computers are not changing at all, as stated by Dourish [9]. Our basic idea about what a computer is, what it does, and how it does it, for instance, has hardly changed for decades. The increase in computational power and the expanding context in which we put that power on, suggest that we need new ways of interacting with computers, ways that are better tuned to our needs and abilities. In the last few years, a new conceptualization of computational phenomena has placed the emphasis not on procedure but on interaction [33]. It emphasizes diversity and specialization rather than unity and generality. Thus, it has affected how we think about computational models of mind, as reflected by Minsky's "Society of Mind" [24], Agre's critique of computational reasoning [1], or Brooks approach to robotics [2].

Human-computer interaction in traditional application development is focused on the interaction between a task and a single user interface designed for a single kind of user. Application user mass is treated as a single entity, making no distinction between the different user stereotypes included in that user mass (figure 1a). A logical evolution should lead interaction to a development model where these stereotypes 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 human-computer interaction design (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 development for a single kind of user, but the real thing is that we are all a little bit different, we might match an user profile, but with our 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.

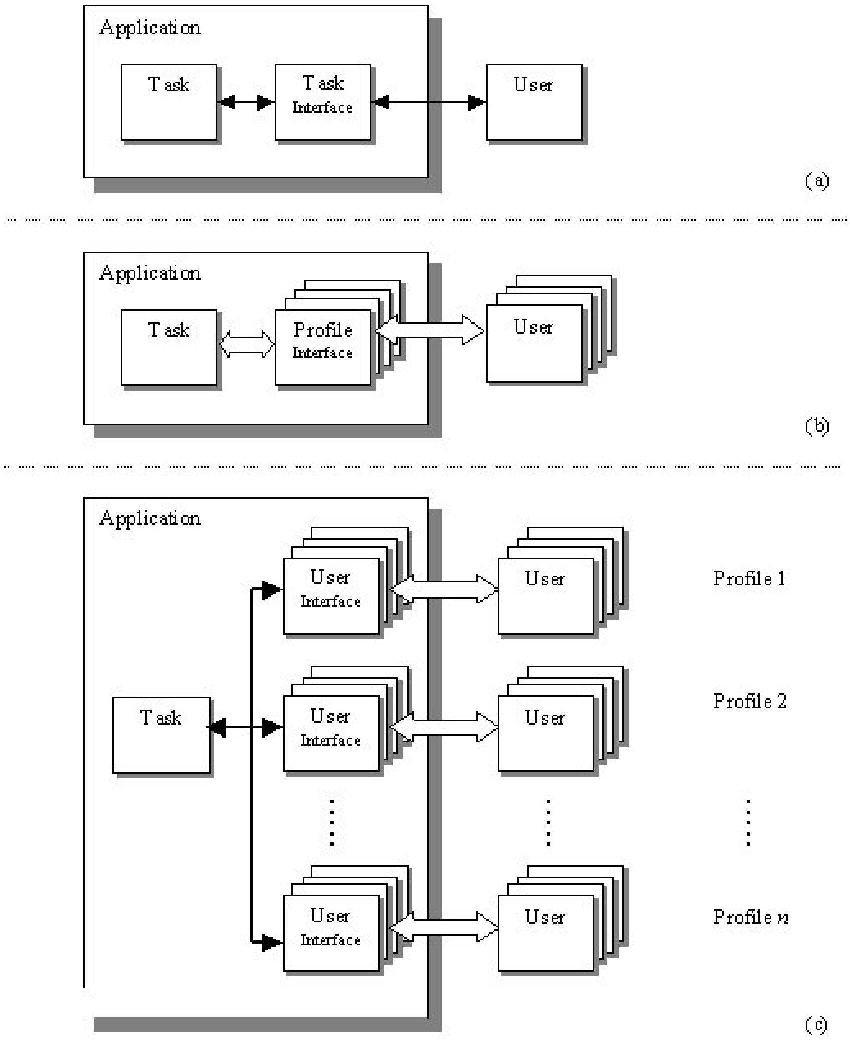


**Fig. 1.** (a) Unity, (b) diversity, and (c) specialization in computer interaction.

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 a user-centered, adaptive human-computer interaction (HCI). As clearly stated by Preece et al [30] "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. [18][27][28][8][3]). Adaptability is also a today's hit [5][31][32][6]. And, there have been some approaches to fit these ideas into concrete architectures [31][10][15][12].

## 2 Social Computing

Social computing refers to the application of sociological understanding to the design of interactive systems. Traditional approaches – based perhaps on functional specifications or on laboratory-based usability studies – tend to be disconnected from the lived detail of the work.

The use of social computing methods is rooted in a distinction between work processes and work practice. Work processes are the formalized or regularized procedures by which work is conducted. It is important to recognize that the duality of practice and process is inevitable. A process description can never eliminate the need to interpret it for specific occasions. Practice is always dynamic, arising as a way to mediate between processes and the circumstances in which they are enacted. The reason to study practice is to understand how this dynamic mediation takes place. The objective of sociology is not to develop abstract theories of social reality, but rather, to understand how social reality is achieved; how people make it work [11]. On the basis of specific observations of activity, social computing attempts to uncover the commonsense methods by which people achieve the orderliness of action.

### 2.1 Accountability

Accountability is an idea essential to social computing in the analysis and development of software systems. The accountable aspect of activity is never a "commentary" on the activity, standing separately from it; rather, it is an intrinsic and inseparable feature of how the activity is woven into the fabric of action and interaction. It is an "endless, ongoing, contingent accomplishment"; the account that matters is one that is good enough for the needs and purposes at hand, in the circumstances in which it arises and for those who are involved in the activity. We pay attention not just to the destination, but also to the route taken to get there. This is critically important in providing a basis for rational mutual action.

Within a system, we know that the different components will interact in fixed and predictable ways. Users are less predictable, though, and their actions less fixed. Users may have different goals in mind, different reasons to and different ways in which they want to use it. The way that activities are organized makes their nature available to others; they can be seen and inspected, observed and reported.

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. So, the account should not simply be an abstract description of the system's behavior, but rather an explanation of how the system's current configuration is a response to the sequence of actions that has led up to this moment, and a step on the path toward completing the larger action in which it is engaged.

Therefore, 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, to reflect the user preferences (figure 2a).

### 2.2 Places

The idea of space is a fundamental aspect of how many interactive systems operate. Things generally appear within the space. There can generally be only one object at a given point in space. Things tend to stay where they have been put. Space, although important, is not the constitutive element of the ways that interactions are organized. Instead, the new view draws a distinction between those interactive phenomena that are derived from the nature of the space in which they unfold, and those that are instead predicated on an understanding of the place that is occupied [16].

By configuring the space in different ways, different kinds of behavior can be supported. While space refers to the physical organization of the environment, place refers to the way that social understandings convey an appropriate behavioral framing for an environment. The important point to recognize here is that these practices emerge not from the designers of the system, but from the actions of its users. This means two things; true places emerge only when really occupied day-to-day, and, place cannot be designed, but only designed for.

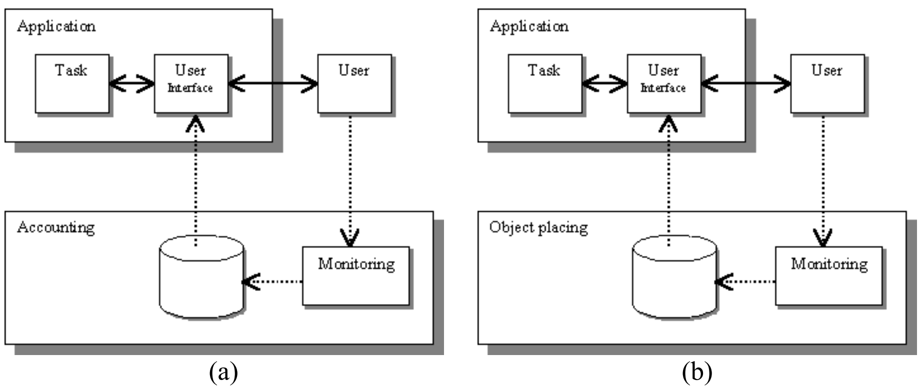


**Fig. 2.** (a) Accountability, and, (b) places in user interaction.

Once again, our proposal is that applications should log the interaction history for the use of places for each user. All these logged information will be gathered and stored in a knowledge base. Thus, the user interface presented to the user could use those data as an input for customizing the user interface presentation, to reflect the user preferences (figure 2b).

## 3 Usability Metrics

Usability metrics are software quality metrics with a long history of successful application in software engineering [4][13][17]. But, metrics also carry risks [7]. No simple number can completely represent anything so subtle and complex as the usability of a software system, but numbers can sometimes create the illusion of understanding. Usability metrics have a number of uses, but mostly from the designer's point of view.

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.

### 3.1 Preference Metrics

One of the most popular ways to assess usability is to use preference metrics. User satisfaction is a component of usability and also an important factor in success in the marketplace. User preferences are generally only relatively weakly correlated with actual ease of use in practice. In some cases, users will prefer designs that are actually harder to use or less efficient.

On good example of a standardized set of preference metrics is the Software usability Measurement Inventory (SUMI) developed as part of the ESPRIT project [29]. SUMI is a 50-item questionnaire that includes five subscales measuring different subjective aspects of usability: affect (how much the user likes the design), efficiency (how well the software enables productive use), helpfulness (how supportive the software and documentation are), control (how consistent and normal the software response is), and learnability (how easy the software is to explore and master). Another approach is the Subjective Usability Scales for Software (SUSS) questionnaire, which measures six key elements of user interface designs affecting usability: valence (liking or personal preference), aesthetics (attractiveness), organization (graphical design and layout), interpretation (understandability), acquisition (ease of learning), and facility (overall ease of use).

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, and even though, they are not so useful when capturing user preferences at run time for intelligent interfaces; they are a key point for design. However, there are some preference metrics, such as the manipulation artifact used when commanding tasks (keyboard, menus, toolbars) that can become especially useful for capturing user preferences (figure 3a).

### 3.2 Performance Metrics

Performance metrics are indices of various aspects of how users perform during actual or simulated work. Measurement studies form the basis of much traditional research on human factors. 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 [26]. Two obvious alternatives exist for measuring the user's performance: either bring some test users into the laboratory and give them a list of the test tasks to perform, or observe a group of users at work at their own environment and measure them whenever a task like the specified test task occurs.

Typical quantifiable usability measurements include: the time users take to complete 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 [26]. 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 user interfaces adapted to user, and leaving behind user interfaces where user must adapt to a given and fixed user interface (figure 3b).

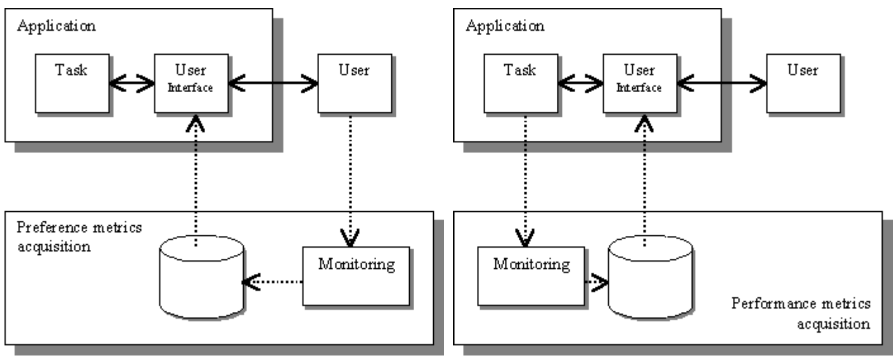


**Fig. 3.** (a) Preference metrics, and, (b) performance metrics in user interaction.

## 4 User-Centered and Adaptive Interaction Agents Architecture

To build a system that enhances the user's motivation in front of his computer, 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 recent study [21] integrates that the person feeling frustrated suddenly has diminished abilities for attention [22], memory retention [20], learning [23], thinking creatively [19], and polite social interaction [14], among other things – as well as a penchant for getting more frustrated in the immediate future. Also, a frustrating episode in interaction with a computer system can leave the user feeling negatively disposed toward the system itself.

Our proposal is to design and build interaction agents that significantly help users in their relationship to computers. 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 [25]. And, the cue is that such interaction agents, which are able to show that one's state is acknowledged and heard, can be built using existing technology.

User-centered design implies the design of systems that are able to interact with their users as social and affective beings, not just as information processors. These kinds of designs engage users in an increased sense of cooperation and good feeling, leading to increased productivity and job satisfaction. The implications span dozens of important domains including human productivity, customer satisfaction, product design, human-computer interaction, human psychology, sociology, and communications – not to mention the quality of day-to-day life for the average computer user.

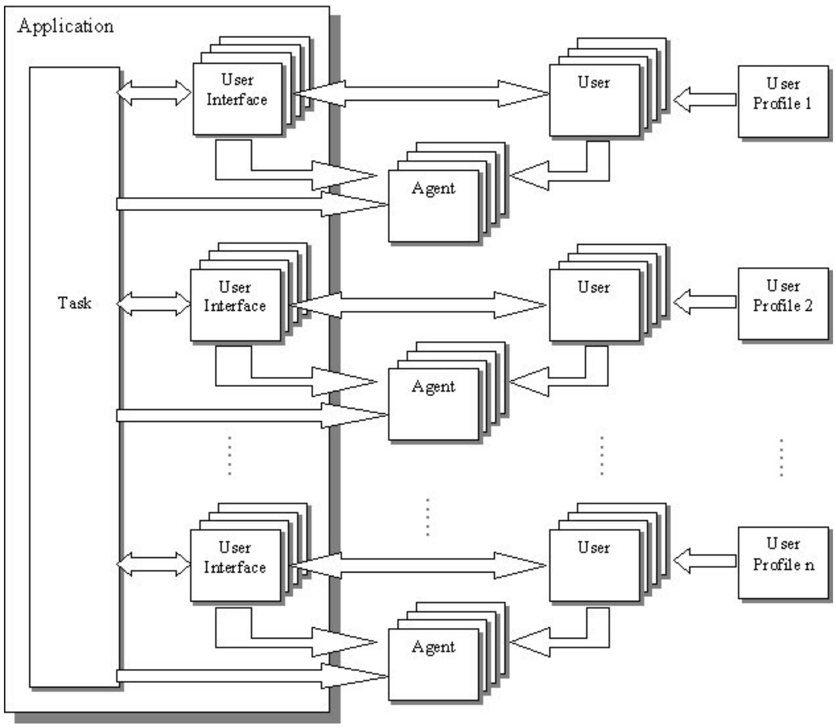


**Fig. 6.** User-centered and adaptive interaction agent architecture.

When engaging in active computer support, the system must actively solicit information about the user's state. This solicitation must be timely. The initiation of the supportive dialogue may come either from the user's own initiative, or the system may proactively initiate the dialogue with the user. This may only is accomplished if the interaction agent may rely on behavioral measures of the user.

The ultimate goal for Human-Computer Interaction 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. Some kind of intelligent agent is required with the ability to capture those changes.

Figure 7 illustrates one possible system architecture for the depicted scenario. In this figure, there are three separated blocks, where a user-centered and adaptive interaction agent has been added to the traditional application-user dialogue. The agent monitors and logs the interaction between the user and the use interface by capturing three different parameters. On one hand, the agent logs the interaction history in an accounting process. On the other hand, the agent tracks the way user interface components are placed, and finally assesses preference metrics, such as the preferred interaction artifact for each action (keyboard, menu, toolbar, etc).

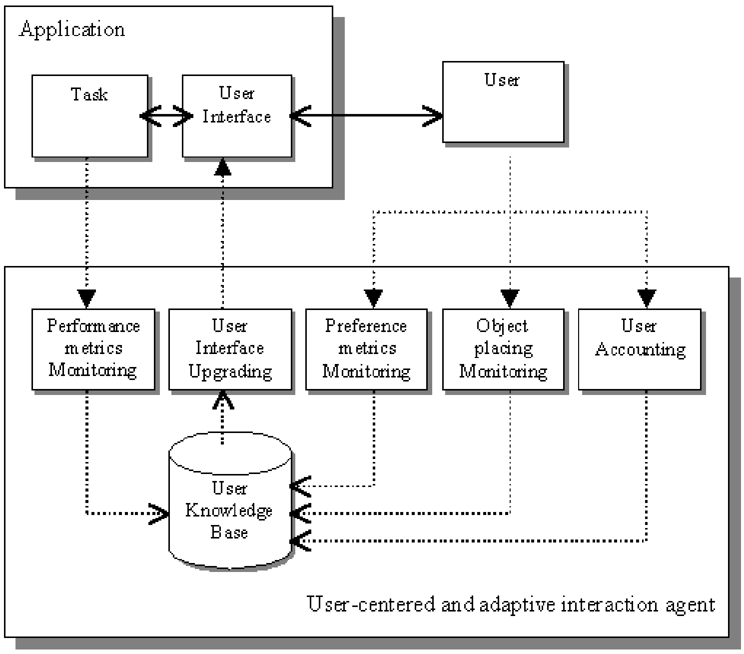


**Fig. 7.** The interaction agent.

Besides the described parameters, the user-centered adaptive interactive agent assesses some behavioral systems issues, 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 has.

## Acknowledgements

This work is supported in part by the Spanish CICYT TIC 2000-1673-C06-06 and CICYT TIC 2000-1106-C02-02 grants.

## References

[1] Agre, P.: Computation and Human Experience. Cambridge University Press (1997)
[2] Brooks, R.: Cambrian Intelligence: The Early History of the New AI. Massachusetts Institute of Technology (1999)
[3] Cañas, J.J., Salmerón, L., Antolí, A., and Fajardo, I.: Propuesta de guía de diseño basada en el modelo mental del usuario. Tercer Congreso Interacción Persona-Ordenador, Interacción 2002 (2002) 3-9
[4] Card, D., and Glass, R: Measuring Software Design Quality. Prentice-Hall (1990)
[5] Cardon, A., and Lesage, F.: Toward adaptive information systems: considering concern and intentionality. Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'98 (1998)
[6] Carro, R.M., Breda, A.M., Castillo, G., and Bajuelos, A.L.: Generación de juegos educativos adaptativos. Tercer Congreso Interacción Persona-Ordenador, Interacción 2002 (2002) 164-171
[7] Constantine.L.L., and Lockwood L.A.D.: Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. Addison-Wesley (1999)
[8] d'Inverno, M., and Luck, M.: Sociological agents for effective social action. Proceedings of the Fourth International Conference on Multi-Agent Systems. IEEE Computer Society (2000) 379-380
[9] Dourish, P.: Where the Action Is: The Foundations of Embodied Interaction. Massachusetts Institute of Technology (2001)
[10] Eisenstein, J., and Rich, C.: Agents and GUIs from task models. International Conference on Intelligent User Interfaces. ACM Press (2002) 47-54
[11] Garfinkel, H.: Studies in Ethnomethodology. Polity Press (1967)
[12] García Serrano, A., Martínez, P., Calle Gómez, J., and Hernández, J.Z.: Propuesta de diseño y evaluación de un modelo de ineracción avanzado. Tercer Congreso Interacción Persona-Ordenador, Interacción 2002 (2002) 76-81
[13] Gilb,T: Software Metrics. Winthrop Publishers, Inc., Cambridge MA (1977)
[14] Goleman, D.: Emotional Intelligence. Bantam Books (1995)

[15] González Rodríguez, M., del Moral Pérez, E., del Puerto Paule Ruiz, M., and Pérez Pérez, J.R.: GADEA: Una arquitectura para el desarrollo de interfaces de usuario adaptables a la diversidad cognitive humana. Tercer Congreso Interacción Persona-Ordenador, Interacción 2002 (2002) 37-42

[16] Harrison, S., and Dourish, P.: Re-place-ing space: The roles of space and place in collaborative systems. Proceedings of the ACM Conference on Computer-Supported Cooperative Work CSCW'96 (1996) 67-76

[17] Henderson-Sellers, B: Object-Oriented Metrics: Measures of Complexity. Prentice-Hall (1996)

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

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

[20] Kahneman, D.: Attention and Effort. Prentice Hall (1973).

[21] Klein, J.T.: Computer Response to User Frustration. Massachusetts Institute of Technology (1999)

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

[23] Lewis V.E., and Williams, R.N.: 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 (1989) 157-171

[24] Minsky, M.: The Society of Mind. Simon & Shuster (1988)

[25] Myers, D.G.: Psychology. Worth Publishers (1989).

[26] Nielssen, J.: Usability Engineering. Academic Press (1993)

[27] Oliveira, E.: Agent's advanced features for negotiation and coordination. In: Luck, M., Marik, V., Stepankova, O., and Trappl, R. (eds.) Multi-Agent Systems and Applications. Lecture Notes in Artificial Intelligence, Vol. 2086. Springer (2001) 173-186

[28] Petta, P. and Trappl, R.: Emotions and agents. In: Luck, M., Marik, V., Stepankova, O., and Trappl, R. (eds.) Multi-Agent Systems and Applications. Lecture Notes in Artificial Intelligence, Vol. 2086. Springer (2001) 301-316

[29] Porteous, M., Kirakowski, J., and Corbett, M: SUMI User Handbook. University College Cork, Ireland (1993)

[30] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., and Carey, T. Human-Computer Interaction. Addison-Wesley (1994)

[31] Sycara, K.P.: Levels of adaptivity in systems of coordinating information systems. In: Klusch, M., and Weiss, G. (eds.) Cooperative Information Agents II. Lecture Notes in Artificial Intelligence, Vol. 1435. Springer (1998) 172-189

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

[33] Wegner, P.: Why interaction is more powerful than algorithms. Communications of the ACM, Vol. 40, No. 5 (1997) 80-91