AGENT-ORIENTED CAPTOLOGY FOR ANTHROPOCENTRIC SYSTEMS

Boldur E. BARBAT

*"Lucian Blaga" University of Sibiu, Faculty of Engineering,
Dept. of Comp. Sc. and Automatic Control, 4 Emil Cioran Street 2400 Sibiu,
ROMANIA Phone: +40-69-212523 Fax: +40-69-212716 E-mail: bbarbat@sibnet.ro*

Abstract: Considering that anthropocentric systems are an adequate domain for captological approaches, and that interface agents are most natural interactants for the humans involved, the paper presents a broad-spectrum generic architectural framework to support developing flexible interfaces for industrial applications, based on synergetic correlation between persuasive technologies and polymorphic agents. The design space for agent-oriented captology is defined and several of its main dimensions are elaborated upon. The main mechanisms used are dynamic priorities, "flexible cloning" and fuzzy temporal windows. Some agent-oriented test-bench applications instantiating the generic architecture, are described briefly.

Keywords: Human-centred design, Man/machine interaction, Agents, Architectures, User interfaces, Medical Applications.

1. INTRODUCTION

Although, regarding awareness, level, intensity and form of expression, they concern the user quite different, both persuasive technologies and agent-orientation (AO) seem to share the feature that, despite being established as research trends in information technology (IT), their significant potential is insufficiently used – at least for anthropocentric systems (AS). Their synergistic connection has been even less investigated: could it become a promising approach, able to diversify the area and methods of human-machine interaction (HCI) in AS?

Why anthropocentric systems? a) Anthropocentric approaches become common in IT, and mainly in industrial systems (Filip, 1999). b) Real time decision support systems have to be anthropocentric (Filip, 1996) so as to encourage all kind of users – from managers to operators – in adopting new working styles and/or acquiring new skills. c) Anthropocentric interfaces are crucial in real-time control – as well as in most other applications involving intensive human-computer interaction (HCI). d) Typically, despite an emerging consensus that context does matter, and the increasing role of human factors in the Internet age, such factors are still undervalued in various environments.

Why captology? First are reasons of effectiveness: a) The obvious role of persuasion in decision making, mainly under time-criticality (Sawaragi and Ogura, 2000). b) The related but distinct impact of credibility (considering both components: trustworthiness and expertise) (Tseng and Fogg, 1999). c) The paramount role of deontological ethics (including its technological and, sometimes, legal implications) (Berdichevsky and Neunschwander, 1999). d) The ramifications of intentionality (setting apart the planned and the side effects). Other reasons stem from IT applied research in AS themselves: e) "The Functional Triad of Computer Persuasion" is distinctly in attendance in industrial applications: the computer can be tool (increasing abilities), medium (providing simulation and virtual environments), and, recently social actor (creating relationships able to persuade people to change behaviours) (Fogg, 1999). f) The prevailing role of modern interfaces in AS (chiefly, multimodal ones). g) The implied anthropocentric approach in HCI.

Why agent-orientation? Since nowadays it is futile to record the motivations of spending intelligence in large-scale systems (LSS), the answers regard the rationale of integrating it in agents or in multi-agent systems (MAS): a) AO manufacturing systems are established and have their design methodologies. b) Internet-based applications – especially in the virtual enterprise (Camarinha-Matos and Pantoja-Lima, 2000) – and many distributed systems require (adaptive) mobile agents to conciliate flexibility and network independence. c) MAS are the natural means to design and implement holonic systems (Fischer, 1998). d) All recent enterprise paradigms (extended, virtual, holonic, fractal, agile, learning, etc.) involve *autonomy* and *empowerment*, essential agentsy features (Guye-Vuillème and Thalmann, 2000). e) Where humans and intentionality are involved (e.g. in anthropocentric systems), the agent is the most natural interactant; f) Almost as a corollary, that means increasing intelligence embodied in and directed to the interface; intelligence in IT is now, above all, made up by agents.

Related work is passed through a threefold filter regarding LSS applications: *flexibility* is high; *robustness* is vital; *response time* is critical (or, at least, clearly limited). Since for *anthropocentric systems*, *agents*, and *mechanisms*, this work was examined in (Barbat, 2000a), here it is only abridged and updated. As Fleming and Cohen (1999) observe, a user-centred approach requires understanding reality: *who* will use the system, *where*, *how*, and to do *what*. Agents under time constraints have to share with humans many concepts (Sawaragi and Ogura, 2000) and improve their negotiation skills. Intentional agents are used more in virtual environments (Martinho and Paiva, 1999). As to the mechanisms, Guarraci (1999) applies polymorphism to facilitate system extensibility and minimize overall software maintenance. Shehory *et al.*, (1999) propose cloning as an approach to the problem of local agent overloads, i.e. to task transfer and agent mobility. *Captology* has had other main targets: besides advertising, the main application areas were promoting and motivating health maintenance, safety behaviour, and environmental conservation (King and Tester, 1999; Fogg, 1999). Even in its few medium-scale applications, persuasion is used rather to change long-range behaviours, than to make time-critical decisions. From the perspective of LSS applications, four are the discernible conclusions: a) Despite being an innovative IT domain, captology is practically not applied in AS. b) Up to now, captology persuaded people to change lasting behaviours. c) Agents under time constraints are sparsely represented in actual systems (perhaps, because of the dissimilar domains they come from, or of their relative novelty). d) It is a disparity between the architecture at the problem-solving level and the structure at the implementation one (thus, neither polymorphism nor cloning are applied in diversifying agent architecture).

The rest of the paper is organised as follows: Section 2 outlines the overall *approach*. The next sections reduce the abstraction level presenting: the *design space for a generic architecture* (Section 3), the *mechanisms at agent level* proposed (Section 4), some *implementation issues* implied by the approach (Section 5). Preliminary *conclusions and intentions* hinted at are closing the paper.

2. APPROACH

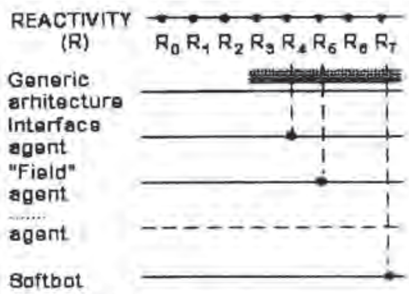
A goal such as filling the niche between AS and captology requires an approach based on "micro-continuity"(a stepwise proceeding is safer and cheaper). Corollaries: a) The approach has to start from all fields considered: *AS*, *agents*, and *captology*. b) A generic architecture for systems integrating them, allows several test applications to be instantiated. c) Existing mechanisms, local approaches, and applications can be reused. Thus, the *premises and criteria* are: a) Whilst at the mechanism and implementation levels a bottom-up approach is serviceable, at conceptual level a top-down approach is preferred. b) Hence, the generic architecture will be based on design spaces for agents to be used in AS. c) In spite of their usefulness in earlier development stages (conceptualisation, design, and so on), all concepts of functional entities, except execution threads, disappear at run-time (the agent partially – insofar its rationale and functionality have been implanted into threads). d) The application instances intended to validate the approach have to: i) allow conclusive incremental testing for both architectonic features, and mechanisms employed; ii) represent solutions to real-world problems – albeit small-scale ones; iii) be easy extendable for further research.

Micro-continuity manifests itself at both the *conceptual* and the *implementation* level: a) A first correlation between persuasive interfaces and intelligent agents, based on a generic architecture, tailored the captological design space for medical informatics; afterwards the space was refined with two dimensions: emotivity and ethical behaviour. Although not in an LSS area, the approach is similar. b) Beside the resultant captological applications two other application categories serve as test bed for the main mechanisms proposed: robotic soccer and agents for web-centred complex knowledge management – see Section 5. The main ideas underlying the undertaking are in fact intertwined, but their core has been outlined separately in (Barbat, 2000a), in a larger framework, regarding the general relationship between holons, agents and threads in AS.

3. DESIGN SPACE FOR GENERIC ARCHITECTURES

Why generic architectures? The answer, considers nested ranges – AS, real-time control, and applications typical for Romania – and is threefold: a) Designing is rather participative, most industrial users being involved from the beginning; functionality is added stepwise. b) The vast range of applications (diversity, scope, requirements, etc.). c) The huge inter-individual variability of all kinds of end-users (e.g., operator, supervisor, engineer, manager), depending on cultural background, previous exposure to IT, motivation, etc., imposes a corresponding interface flexibility. All general reasons for other facets of flexibility, are valid too. Oversimplifying, a generic architecture is the common denominator of all specific architectures it is able to originate. After instantiating it, all architecture instances – from modelling to real-time monitoring, or from telework to enterprise data bases – are autonomous, becoming impossible to be aware of the kernel they are coming from. At the interface level the application has to *be adapted* by the users and to *adapt itself* to them. While *adaptability* (exogenous: adapting system behaviour to user explicit requests) can be attained easily through generic, parameterised functions, *adaptivity* (endogenous: learning to shape, refine, and update behaviour so as to meet implicit user expectations) involves distributed intelligence. Thus, the generic architecture must be AO.

Shaping captological or agentsy requirements for AS, means to separate the relevant concerns, i.e. to identify design-space dimensions and to delimit aspect segments. (Concepts as "design space", "concern", "aspect" etc. have the same meaning as in generative programming, where they come from.) Since the design space for agents has been defined earlier for medical applications (Barbat, 1999), and for information retrieval systems (Barbat and Zamfirescu, 1999), so as to impair redundancy, here just one example, describing an agent design-space dimension present in all applications, but key element in LSS: *reactivity* (i.e., action in response to stimuli). Agents perceive their environment and respond appropriate to changes that occur in it, treated as stimuli (at architectonic level) or as events (at implementation level). Environments are very diverse (the examples relate to real-time control): physical world (in industrial interfaces), user (everywhere, but essential in AS), data warehouses (in data mining), the World Wide Web (in virtual enterprises), habitually some mixture of all these. Reactivity is decisive in real-time control (to ensure robustness in non-deterministic environments), but no agent can lack it, to be able to satisfy its owner's requests; so as to react promptly, the agents are interrupt-driven. Thus, its aspects range from handling simple static input, to refining ontologies with soft computing techniques (see Figure 1).



Aspects

R0 Static input
R1 Interactive input
R2 Procedures/Functions
R3 Exception handling
R4 Transaction handling
R5 Subsumption architecture
R6 Artificial Neural Networks
R7 Soft computing

Fig. 1. Dimension aspects for reactivity

Likewise, since the design space for agents has been adapted to medical captology (Barbat *et al.*, 2000) and to real-time control (highlighting its specificity for AS) (Barbat, 2000a), here will be focused only upon the differences between captological design spaces for AS and those for medical informatics: a) *Levels of analysis*. What user population is aimed at? Although the individual level remains focal – at least for interface agents – the organizational level becomes crucial for implementing the new enterprise paradigms. b) *Credibility*. Under time-criticality, it has a key impact. From the four suggested types of credibility (Tseng and Fogg, 1999) the *presumed* (based on assumptions and stereotypes) and the *reputed* (based on third-party reports) are too dangerous, and the *surface* (based on simply, almost reflex inspection) can be risky; thus, only *experienced* credibility (based on first-hand experience) seems safe. c) *Persuasive strategies*. For LSS, among the basic strategies (positive or negative feedback, role playing, simulated experiences, surveillance, environments of discovery, virtual groups, personalizing, (King and Tester, 1999)), those supporting cooperative work and group decision making will dominate. d) *Form factors*. Beside the desktop-based systems (from personal computers to specialized ones), the other two categories, i.e. artefact-based (portable, embedded, eudemonic, etc.) and environment-based (integrated into the user setting, distributed) will play a main role. e) *User profile*. The huge inter-individual diversity displayed by all kinds of users (e.g., depending on personality, cultural background, profession, area of interest, social status, motivation, level of information, previous exposure to IT) is still amplified under time-criticality (stress alone is enough to magnify behaviour dissimilarities).

The only added dimension is *time horizon*. Indeed, considering the vast diversity of LSS, persuasion cannot be the same in every context. Two examples: i) Although the aspect of "being open to inspection" is crucial for both credibility and ethical reasons, in time-critical applications it must be abridged or avoided altogether. The agent must persuade first and explain later (the user, simply, has no time to check its claims). ii) The adequate psychological instruments are different (e.g., the message colours have much greater impact if the user must make instantly a decision than if the aim is to quit smoking). Maybe, the aspects should range from seconds to years for contexts as diverse as nuclear plants and lifelong learning (see next Section). In the end, the design space, S'', becomes a hypercube with amalgamated captological and agentsy dimensions (some examples are given in Figure 2).

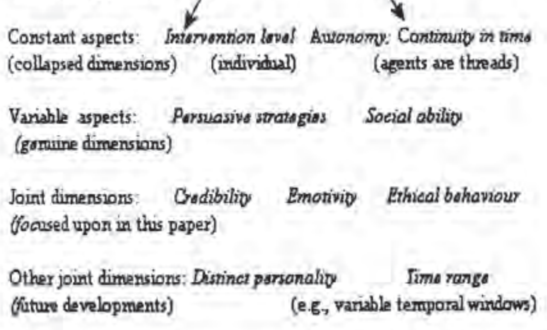


Fig 2. Design-space dimensions of captological agents in anthropocentric systems

4. AGENT-LEVEL MECHANISMS

Agents are supposed to represent humans and act for them (this is the core of the metaphor). Thus, they have to be autonomous and lasting. Corollary: agents are tasks or threads and all agent-level mechanisms must be embodied into threads. Due to this lower abstraction level, the same mechanisms can be applied to different agent types. On the other hand, new architectonic features may need new mechanisms. Thus, from the perspective of captological agents for AS, the required mechanisms fall into three categories: A) already applied for captological agents in medical applications. B) Rather general ones, useful for any agents in AS. C) Specific. (The first two categories, being implemented and already described previously, will be only reviewed by means of an example.)

A) *Clone-based polymorphism*. In nontrivial applications, so as to be credible, agents have to be lifelike, i.e. pathematic (Barbat, 2001). The emotional exteriorisation shows a vast variability implying many "emotional variables" to be handled at run time (facial expression, voice, movement, etc.); thus, the agent must be polymorphic. This kind of polymorphism is implemented through flexible cloning. (In the application described in (Barbat, 2001), after noticing that the patient lights a cigarette, the pathematic agent talks with a throaty voice. However, its voice is not altered: instead, the agent is cloned, avoiding a "voice" variable as well as the code to modify it.)

B) *Priority-driven agents*. Priorities are often changed dynamically so that they become one of the best tools to fine-tune agent behaviour. Dynamic priorities for threads embodying agent behaviour allow implementing fine granularity in monitoring their actions at all architectural levels. (In the application mentioned above, the agents' reactions slide down in time. Since this mechanism is not implemented in the operating system used, a "dynamic priority decrement"-mechanism was simulated. Thus, an emotion can be overridden by new emotions – the old one is losing its supremacy because the agent thread has a decreasing priority.)

C) *Fuzzy temporal windows.* In order to improve the obvious drawbacks of the solution described above, a new mechanism is now tested. As stated by Coudert *et al.* (2000), "the idea of modelling relaxable time constraints by fuzzy sets" has been used in scheduling manufacturing and maintenance operations. Using also techniques applied in fuzzy controllers, the mechanism is slightly modified so as to match the clear-cut asymmetry of the rise and fall of agent emotions. In fact, quite similar variants of the window profile can be easily applied to many aspects of the time-horizon dimension. Some examples of application areas, arranged along with increasing time periods available for persuasion: space craft, process control, group decision making, maintenance planning, business process reengineering, long-range strategies.

5. IMPLEMENTATION ISSUES

To be confirmed, such a shift in approaching the role of interface agents in anthropocentric LSS requires an adequate test bench, one where the threats are major and the challenge for threads to cope with is salient too. However, technological, economic, workforce, and time constraints imposed to reach results quickly. Thus, implementation started with applications that had to be: *diverse* (to cover all new aspects and to test the concepts and mechanisms one by one), *simple* (to develop them fast), but intrinsic *usable* (to be at least roughly conclusive), and, if possible, *underway* (to avoid starting from "tabula rasa").

Especially such key concerns as persuasion, polymorphism, or cloning can be investigated easier in a less complex and more permissive setting of agents – not even MAS, just a few agents to start with. Therefore, innovative areas like medical captology offer a valuable potential for both exploring mechanisms and checking architectures. Hence, each conceptual step (e.g., a new design-space dimension) was paired by an application instance. The main stages were: a) An AO generic architecture for medical applications, instantiated through an multi-modal persuasive interface (Barbat, 1999). b) The approach was extended to AO captology for medical informatics, with an improved version of the same application (Barbat *et al.*, 2000). c) A generic architecture was defined for pathematic agents and instantiated by an application with several "credible" agents (Barbat, 2001; Bichis *et al.*, 2001) . d) A similar scheme has been applied for agents exhibiting various ethical attitudes (Barbat, 2000b; Bichis *et al.*, 2000). At the same time, two other application types, were used as test bed for some of the mechanisms described above (details are given in (Barbat, 2000a)): a) *Agents for Web-centred complex knowledge management*. Several search-engine agents are used in the attempt to line out a generic polymorphic profiler agent. Its instances will be applied to build, refine and update various user, as well as search-engine, profiles; a first such agent, still under development, is intended to model e-business users (Barbat and Zamfirescu, 2001). b) *Robotic soccer*. Rationale: is a genuine real time application, widespread employed, emblematic for MAS, allowing a multi-agent approach to holonic systems. As regards testing, it is an excellent playing (and battle) field for concepts and mechanisms. Thus, any change can be easily evaluated, simultaneously with its side effects: the team with the change inoculated, plays against the old one; the outcome is the judge (Cândea *et al.*, 2000).

Since most implementation issues have been presented in (Barbat, 2000a) here are summarised only those relevant to the captological applications: a) *Operational environments*. The underlying system is a networked high-end PC (with Windows 2000). The agents are written in Visual C++ 6.0 and use Internet Explorer 5.0 because it has integrated parsers for HTML and XML. (Java was avoided because of the the smoke detector device – communicating through a serial port). The captological agents are based on the Microsoft Agent technology. b) *Multi-modal interfaces*. The framework applied supports a full range of multimedia components to be embedded. Animated characters appear directly on the desktop as interactive assistants. Characters interact with the user, using synthesized speech (Text-To-Speech Engine) and text in cartoon word balloons. c) *Active Desktop*. Some agents interact more closely with the operating system shell, enabling the "Active Desktop" feature; this is used also to publish on the desktop a web page with subliminal messages. d) *Authentication*. Because of the test-bench nature of most applications, there are several user categories. The right to privacy, imposed a scheme of user groups, according to distinct profiles and levels, expressed through privileges/rights/restriction filters.

6. CONCLUSIONS AND INTENTIONS

The conclusions can be only partial ones, because a new approach cannot be evaluated after a few instantiations. The implementations are more than a surrogate, but less than a groundwork for ecological validation. Nevertheless, the approach is promising because: a) Captology is here to stay, due to the obvious role of persuasion in time-critical decision-making as well as in e-Business (especially for negotiation and advertising). b) In the innovative context of captological medical applications, persuasive agents can contribute to both therapy effectiveness and patient acceptance. Hence, comparable results can be expected for AS, where the operator-agent interaction is basically similar. c) The vigorous expansion of intentional agents in AS comprises already inconspicuous persuasive technologies. Thus, it seems better to handle them explicit and systematic. d) Agent-oriented generic architectures based on design-space dimensions confirm their utility also in captology for both research and application development. e) All application instances endorse the generic architecture they originate from. f) The applications are operational and developing them as instances of a generic architecture was straightforward and trouble-free. g) Although the applications were not intended to become commercial ones, the framework can be used for commercial applications. h) The new mechanisms proposed work, but since their efficiency is highly application-dependent, only large-scale experiments could give resolute answers. (Clone-based polymorphism worked well in developing agents and seems to have a clear potential as an alternative to polymorphic operators in object-oriented approaches).

From a general perspective, the future research objectives are: a) Extending the approach to other captology areas (e.g., e-commerce) and to perceptual interfaces. b) Investigating the impact that dependability of the conveyed information (approximate, fuzzy, uncertain) can have on persuasion and on agent behaviour. c) Setting up a "polymorphic agent factory", based on extensive cloning. d) Testing the approach via a challenging real-time application (e.g., in nuclear technology), able to validate it.

ACKNOWLEDGEMENTS. This work was partially supported by ISW Sibiu.

REFERENCES

Barbat, B. (1999). Agent-Oriented Architectures for Medical Applications. MEDINF99 Conference, CD-ROM, Sibiu.

Barbat, B. and C. B. Zamfirescu (1999). A^3CKM: Anthropocentric Agent Architectures for complex knowledge management. *Acta Universitatis Cibiniensis*, **XXXVIII**, (E.), 23-28, Sibiu.

Barbat, B., C.B. Zamfirescu, and G. Costache (2000). Agent-Oriented Captology For Medical Informatics. In: *Medical Infobahn for Europe.* Proceedings *of MIE2000 and GMDS2000* (A. Hasman *et al.*, Eds.) pp. 465-469 IOS Press, Amsterdam.

Barbat, B. (2000a). Holons, Agents, and Threads in Anthropocentric Systems. Studies in Informatics and Control Journal, **9**, (3), 253-268.

Barbat, B. (2000b). Agents and Ethics in Medical Captology. *MEDINF2000 Conf.* Iasi (in print).

Barbat, B. (2001). Pathematic Agents for Medical Informatics. *Proc. Int. NAISO Congress,* ENAIS (in print).

Barbat, B. and C. B. Zamfirescu (2001). Polymorphic Agents for Modelling E-Business Users. *Proc. Int. NAISO Congress* EBAA (in print).

Berdichevsky, D. and E. Neunschwander (1999). Toward an Ethics of Persuasive Technologies. *Comm. ACM*, **42** (5), 51-58.

Bichis, C., C. Stefan, D. Pipernea and B. Barbat (2001). Developing Credible Agents for Captology. *Proc. Int. NAISO Congress* ENAIS (in print).

Bichis, C., C. Stefan, D. Pipernea, and B. Barbat (2000). Implementing Ethical Aspects in Agents for Medical Applications, MEDINF2000 Conf. Iasi (in print).

Camarinha-Matos, and C. Pantoja-Lima (2000). Supporting Business Process Management and Coordination in a Virtual Enterprise. In: *Advances in networked enterprises. BASYS 2000, 4th IFIP/IEEE Int. Conf.*, (L. Camarinha-Matos *et al.*, Eds.), pp. 3-14. Kluwer Academic Publishers, Boston.

Cândea, C., M. Staicu and B. Barbat (2000). Holon-Like Approach for Robotic Soccer. *Proc. EuroRoboCup 2000,* CD-ROM, Amsterdam.

Coudert, T., Grabot, B., Archimède, B. (2000). Integration of Maintenance Constraints in Scheduling: Fuzzy Modelling and Multi-Agent Approach. In: *Advances in networked enterprises. BASYS 2000, 4th IFIP/IEEE Int. Conf.* (L. Camarinha-Matos *et al.*, Eds.), pp. 297-304. Kluwer Academic Publishers, Boston.

Filip, F. G. (1996). DSS for enterprise and management reengineering: towards anthropocentric systems. In: *Proc. CESA 96 IMACS Multiconf., Symp. "Modelling, Analysis & Simulation"* (P. Borne, Ed.), vol. 1, pp. 438-443,.

Filip, F. G. (1999). Information Systems: from Job Aids to Intelligent Assistants. Acta Universitatis Cibiniensis, **XXXVIII**, (E.), 1-9, Sibiu.

Fischer, K., (1998). An Agent-Based Approach to Holonic Manufacturing Systems. In: *Intelligent Systems for Manufacturing. Multi-Agent Systems and Virtual Organisations* (L. M. Camarinha-Matos *et al.*, Eds.) pp. 3-12. Kluwer Academic Publishers, Boston.

Fleming, M. and R. Cohen (1999). User Modeling in the Design of Interactive Interface Agents.In: Proc. of the Seventh International Conference on User Modeling (J. Kay Ed.), pp. 67-76. Springer, New York.

Fogg, B. J. (1999). Persuasive Technologies. *Comm. ACM*, **42** (5), 27-29.

Guarraci, B. (1999). Using polymorphism to create complex agents. *Proc. of the third Annual Conf. on Autonomous Agents*, ACM Press.

Guye-Vuillème, A. and D. Thalmann (2000). Requirements for an architecture for believable social agents. *Proc. of the 4th Int. Conf. on Autonomous agents*, 48 – 49, Barcelona.

King, P. and J. Tester (1999). The Landscape of Persuasive Technologies. *Comm. ACM*, **42** (5), 31-38.

Martinho, C. and A. Paiva (1999). Pathematic Agents: Rapid Development of Believable Emotional Agents in Intelligent Virtual Environments. *Proc. of Autonomous Agents99 Conf.,* ACM Press.

Sawaragi, T. and T. Ogura (2000). Concept Sharing Between Human and Interface Agent Under Time Criticality. In: *Advances in networked enterprises. BASYS 2000, 4th IFIP/IEEE Int. Conf.* (L. Camarinha-Matos *et al.*, Eds.), pp. 269-278. Kluwer Academic Publishers, Boston.

Shehory, O., K. Sycara, P. Chalasani and S. Jha (1999). Agent Cloning: an approach to agent mobility and resource allocation. The Robotics Institute. Carnegie Mellon University, Pittsburgh USA.

Tseng, S. and B. J. Fogg (1999). Credibility and Computing Technology. *Comm. ACM*, **42** (5), 39-44.