Combining amorphous computing and reactive agent-based systems: a paradigm for pervasive intelligence?

David Servat
MIRIAD/OASIS/LIP6, Université de Paris 6
Boîte 169, 4 Place Jussieu
75252 Paris Cedex 05 - France
David.Servat@lip6.fr

Alexis Drogoul
MIRIAD/OASIS/LIP6, Université de Paris 6
Boîte 169, 4 Place Jussieu
75252 Paris Cedex 05 - France
Alexis.Drogoul@lip6.fr

ABSTRACT

This paper is intended to provide insight on the upheaval that the advent of computing systems based on pervasive computing will represent. We show, after an introduction to this concept, why the traditional rules of design, validation, control and deployment nowadays used in the industry are not adapted anymore to these new systems. We then show that the multi-agent systems domain seems, in a number of ways, to be able to answer some of the problems they raise. But we also claim that designing these "new" MAS will certainly require us to consider alternative sources of inspiration, other than economy or sociology. One promising direction, which we call Pervasive Intelligence, PI, is to view and design them as "ecosystems of physical agents" (heterogeneous, open, dynamic, and massive group of interacting agents), organized after "biological", "physical" or "chemical" principles. Two domains of research illustrate this direction: reactive multi-agent systems and amorphous computing. The first one studies complex, self-organized, situated systems that rely on biological metaphors of communication and organization. The second one, which draws its inspiration from chemistry, tries to develop engineering principles to obtain coherent behavior from the cooperation of large numbers of unreliable computing parts connected in irregular and time-varying ways. We think that a convergence between the researches conducted in these two domains would pave the way towards suitable paradigms and methodologies to tackle the challenge that Pervasive Intelligence represents for the next decade.

Keywords

pervasive computing, amorphous computing, reactive MAS, emergence, self-organization

1. INTRODUCTION: FACTS ON PERVASIVE COMPUTING

In 1991, Mark Weiser, from Xerox Parc, following the same path that was initiated by Licklider [32] in the early 1960s, introduced in a well-known paper [54] the concept of "ubiquitous computing", more commonly known today as pervasive computing [37]. This concept embraces the advent of new computing systems, consisting of smart communicating devices, which we are likely to be more and more surrounded with in our working places, at home and during our leisure. There are already early signs of this digital invasion, for instance the increasing number of microchips embedded in the objects we use everyday. It has been shown [55] that an adult living in today's industrialized countries is confronted with an average of 40 microchips a day, among which 5 are able to communicate in networks. It is estimated that within 10 years we will be in permanent contact with hundreds of such chips, the majority of which will access dense information networks. Most of these artefacts take the form of everyday life objects (e.g. tools, domestic appliances, clothes, etc.) that are "augmented" with sensors, actuators, processors, and embedded software, so as to communicate or process information, thus enhancing the comfort of use or providing support for new services. Some will be mobile, either on their own, or because they will be worn by their users (phones, clothes, glasses [33], etc.), driven by them (cars), or even be part of them (prosthesis). Others will be static and invisible, integrated in thousands within "smart materials". Most of them will eventually be conceived with means of being aware of certain properties of their immediate environment [40], including other devices, their users and various information sources, so as to be able to function in a coherent, seamless and adaptive way.

What is new in these systems is that most of them will be granted communication capabilities (either local via wireless technologies, or at a distance via networking infrastructures). Fundamentally open, these systems will enable new forms of interaction, as well as online evolution or reconfiguration. The artefacts directly in contact with users will benefit from intuitive interface (natural language or gestual interface, see for instance [51] [52]), and will be built so as to seamlessly integrate in communities, offering support for new collaborative work and as such, will drastically change our way to communicate, work and perhaps, live.

The main reason for the emergence of these systems is economic. The computer industries, hit by the early signs of recession on their traditional markets, franticly seek new prospects to match the steady increase of their production capacities: the moodiness of the traditional vector of computerization (the computer) calls for a diversification of supports which may embed chips and softwares. This is already the case in multimedia companies. Besides, this evolution is acknowledged by famous laboratories (MediaLab, Xerox Parc, etc.) and financial institutions (NSF, Esprit, etc.). This means that sooner or later, we will live among the kind of artefacts we defined above, even if most of the needs they are supposed to fill are still unknown. The analysis of the adoption process of today's technological devices yet shows that the law of offer and demand does inverse more often than not: the introduction on market of new devices results in new or alters the established habits (thus creating a self-satisfying demand). The cellular phone is a typical example, which grew within two years from social status object to mass-produced device, and changed our communication practices, leading to different forms of social relationship that we are just about to analyze.

Pervasive computing happens to be more than just a "gadgetization" craze: serious applications are already studied in the educational domain for instance (tele-education, interaction among students [4], [3]), in the health domain (mobile monitoring systems [48]), in transports (smart cars [24] and roads [21]), and commerce (see [45] on supermarkets), as well as in the military domain (most of the research done on "wearable computers" [41] is granted by DARPA in the United States). These are a few examples, of course, among the numerous existing projects (see [53]), but what they tell us, is what our future is likely to be.

2. CONSEQUENCES OF THE EMERGENCE OF PERVASIVE COMPUTING SYSTEMS

This mutation is not a revolution. It is on the contrary a logical consequence of the evolution of the relationships that users have had towards computing systems, which has been characterized both by a democratization of the access to equipments and a decentralization of the underlying infrastructure. The first period (1950-1970) involved resource sharing among users operating behind central terminals. In the 1980s the success of the personal computer kicked off the history of user-personal computer relationship. In the 1990s both these conceptions eventually merged into Internet technology, an infrastructure enabling resource sharing via personal computer. Indeed Internet represents nothing more than one step towards the advent of pervasive computing sytems: the same logic of simplification and decentralization is prevailing today, and shall lead to a situation where hundreds of computing devices will share us via networks, only far more intricate than current Internet.

These devices will be the components of a computing infrastructure that will radically differ from the ones we know today. As a matter of fact, these systems will:

- not be possibly centrally controlled, either because it will be impractical, or because different people (users, owners, designers, etc.) will control their components,
- see their configuration vary over time, due to the dynamic introduction or deletion of components, or because of changes in the way people use and interact with them,
- be immersed in human collectivities with various sizes and needs (not just one user at a time), and will operate with incomplete information about this social environment,
- federate highly heterogeneous combinations of software and hardware, which will differ by their function or their processing, communication and action capabilities,
- be the result of combinations of components that could not have been foreseen at design time but nevertheless produce (possibly) interesting emergent behaviors,
- continuously need to adapt to their (social, physical and computing) environment in order to improve their efficiency.

These characteristics raise new questions for designers:

- design: which techniques and methods will enable the design and control of such systems under such unpredictable constraints?
- organization: how these artefacts will be able to recognize themselves, coordinate themselves and under what conditions? What kind of knowledge will they need to favor synergies and solve conflicts?
- usage: it is likely that the installation of such systems will deeply affect users' habits. To what extent users will trust such decentralized systems? How will they intervene on the configuration of such systems to tune them to their needs? What impact such systems will have on the organizations in which they will be immersed?
- adaptation: how will these artefacts adapt to the environment in which they will evolve? How will they form collective functional conformations and track such conformations in memory?
- deployment: what kind of protocols of interaction and coordination, what organizational modes will lead to sufficiently safe, robust systems, still flexible to use in such highly variable environments?

2.1 Current research

Put it briefly, most of today's research deal with forecasted hardware problems that such systems are likely to raise: installation of necessary infrastructures, technical feasability, conception of artefacts, with an emphasis on networking and distributed aspects (development of communication protocols, such as Bluetooth(tm) [6] or IEEE 802.11b norm [25], reflexions on the "embedded internet" [8], or even the deployment of IPv6 [26]). Software engineering is playing an important role in the development of virtual byte-code machines and mobile codes, but also with the conception of middlewares (e.g. Jini(tm) [49]), which enable the interoperability among heterogeneous artefacts. The creation of artefacts gather a lot of works on interfaces (haptic interfaces [52], etc.), co-building [56], energetic autonomy, industrial design, miniaturization of components (nano-machines, molecular computers), sensor and actuator conception, etc.

There are already projects which investigate the questions of the application of such systems on a real scale, and the role they will play in everyday life. Among these works, some deal with educational aspects (see for instance [42] or [34]). Smart environments have already been tested in museums (robots guides [50], augmented reality [46]) or in some companies. However so far, these studies lack interesting scenarios and only prove how puzzled designers are in general with what relevant technologies will finally prevail. Most of those scenarios consist of crude copy of the paradigmatic example given by Weiser [54]: "Sal awakens: she smells coffee. A few minutes ago her alarm clock, alerted by her restless rolling before waking, had quietly asked "coffee?", and she had mumbled "yes." "Yes" and "no" are the only words it knows."

2.2 Building collective autonomous smart systems

Between those two axes, most remain to be done, on top of which such acute question as how to control those systems. Indeed, though a lot of energy has been dedicated to standardization, the diversity and high rate of renewal of hardware and software truly invalid the belief if any in a central organization for these systems. On the contrary they need to be conceived right from the beginning as decentralized, and granted smart social abilities (through the use of software executing on them), whose complexity depends on their means of action and perception on their environment. What is at stake is our ability to design the bases for self-organizing means of control. This calls for the adoption of two simultaneoulsy important principles:

1. think of the means for these artefacts to build their own collective structures, with homeostatic properties (e.g. reconfiguration, adaptation) and little human intervention;
2. figure out means for these systems, self-organized as they are, to still produce predictable behaviors, meeting the constraints that users prescribe.

Their distributed nature, their heterogeneity and mobility will make these systems more complex than are today's systems, much so in fact as natural systems are. These issues are not likely to be addressed in a simple way and will require the development of new computing paradigms. Most of them will however derive from the concepts, tools, and methods developed in the field of DAI. This domain is one of the few to naturally deal with the "decentralized mindset" required for apprehending these systems [47]. However, this will also require to rethink many of its underlying models.

Designing these "new" MAS will certainly require us to consider alternative sources of inspiration, other than economy or sociology. One promising direction (which we call Pervasive Intelligence, PI) is to view and design them as "ecosystems of physical agents" (heterogeneous, open, dynamic, and massive group of interacting agents), organized after "biological", "physical" or "chemical" principles. Two domains of research illustrate this direction: reactive multi-agent systems [14] and amorphous computing [13]. The first one studies complex, self-organized, situated systems that rely on biological metaphors of communication and organization [39]. The second one, which draws its inspiration from chemistry, tries to develop engineering principles to obtain coherent behavior from the cooperation of large numbers of unreliable computing parts connected in irregular and time-varying ways [44]. PI is an ideal combination of these two domains, synthesized in the following question: "How to obtain coherent and predictable behaviors from the cooperation of large numbers of situated, heterogeneous agents that are interacting in unknown, irregular, and time-varying ways?". Answering this question would allow us to design truly intelligent, adaptive and autonomous pervasive computing systems.

3. PARADIGMS FOR PERVASIVE INTELLIGENCE

3.1 Shortcomings of present DAI

The current trends in DAI do not seem to our mind to lead to means of apprehending these complex systems as naturally as they could be. The influence of both cognitivist AI and software engineering (noticeable in the orientation of the FIPA works [23]) result in the multiplication of "industrially correct" systems (verified, validated), which, oddly enough, do not seem to be well adapted to the new industrial deal that is taking place. If we consider as references the papers recently published in the proceedings of ICMAS [20], MAAMAW [12] or in AAMAS journal [28], it is quite clear that the majority of the works done on MAS do not show suitable properties as far as Pervasive Intelligence is concerned. Indeed, most of the (theoretical or empirical) works on multi-agent systems rely on:

- small populations of coarse-grained agents [27], while these new systems will probably exhibit massive sets of fine-grained entities, executing on less demanding hardware support (cellular phones, etc.),
- reliable and standardized communication [22], while communication among these systems will undoubtedly have to be fault-tolerant and adaptive to cope with time-varying environments,
- homogeneous architectures (for instance BDI [43]), while incorporating heterogeneity into the design process will be a real necessity,
- closed and static environments, while these systems will behave in open and highly dynamic ones,
- deterministic organizational schemes, while there will be a need for more flexibility, including the emergence of collective properties,
- a deliberate belief [27] to give formal grounds to design in the hope that it will enable the validation of all or parts of the collective execution,
- "ethereal" agents, while most of them will be embodied and situated [16].

Obviously, these choices are not all to be rejected and the hetoregeneity of PI systems itself call for the use of different paradigms, ranging from rational to reactive. Still a certain revision of statements which seem more dogmatic than pragmatic might be needed, especially as far as the validation process and formal specification are concerned.

3.2 Reactive MAS

On the contrary reactive MAS exhibit interesting features that put them as straight-forward candidates for PI systems [15], among them:

- their design principles often rely on metaphors, most of which are well adapted to systems consisting of heterogeneous agents (e.g. ecosystems),
- reactive agents are not necessarily granted rational behaviors and often rely on light-weight individual architectures,
- they make use of massive population of agents and cope with poor communication and probabilistic interactions,
- they rely on self-organization principles instead of fixed organizational schemes,
- the environment itself in which agents act is considered as dynamic and does not generally feature any symbolic representation,
- validation of reactive systems is primarily empirical and adapt well to "situated" approaches (e.g. robotics),
- they naturally embed evolutionist methods or reinforcement learning techniques [19].

Yet this relative straight-forwardness is somewhat outweighed by the difficulty faced when designing such systems. Achieving relevant collective behaviors for instance requires considerable experimental work which may not always come up with reusable methods for other application domains. The set of generic principles and methods amounts so far either to computational models (e.g. eco-resolution [15]) or communication models (e.g. through the environment), rather than provides design methodologies. In spite of the huge amount of research in this field, there is apparently still a long way off before we might be able to construct from scratch "ecosystems of physical agents" with biologically-inspired, organizational principles.

3.3 Amorphous computing

Amorphous computing [5] could fill this gap. This domain ambitions to propose a new paradigm of design and development for a particular subdomain of Pervasive Intelligence: nano-machines and cellular computers [30]. This field was born from the impulse of physicists, mathematicians, computer scientists and biologists concerned by software engineering problems to obtain coherent behavior from non trustable computing devices interconnected in local, unknown, irregular and time-varying ways [13]. The main ideas are drawn from physics and current projects (see [44] for instance) give promising insight on putative design methodologies enabling the building or even self-building of collective systems. To our mind this would provide interesting bases for reactive MAS.

"A colony of cells cooperates to form a multicellular organism under the direction of a genetic program shared by the members of the colony. A swarm of bees cooperates to construct a hive. Humans group together to build towns, cities, and nations. These examples raise fundamental questions for the organization of computing systems:

- How do we obtain coherent behavior from the cooperation of large numbers of unreliable parts that are interconnected in unknown, irregular, and time-varying ways?
- What are the methods for instructing myriads of programmable entities to cooperate to achieve particular goals?

These questions have been recognized as fundamental for generations. Now is an opportune time to tackle the engineering of emergent order: to identify the engineering principles and languages that can be used to observe, control, organize, and exploit the behavior of programmable multitudes.

We call this effort the study of amorphous computing." excerpt from the MIT Amorphous computing project Web page [5].

Putting aside the ambitious tone, what is interesting in this "profession of faith" is the deliberate engineering stance towards self-organization and emergence phenomena. The motives are less the understanding of how things work than how to make things work. It reminds us of the "intentional stance" promoted by Dennet which provided an interesting framework to analyze distrubuted systems. This engineering perspective is backed up by a strict framework of investigation: the amorphous computer model, which is a computable abstraction of what could be an amorphous medium.

3.3.1 *The amorphous computer model*

The basic assumptions come from the forecast that when programmable entities (called processors or particles in amorphous computing literature) will be embedded in materials they will have to be mass-produced, hence offering very low guarantee on their effective functioning [2].

An amorphous computing medium is a system of irregularly placed, asynchronous, locally interacting computing elements: for instance one can imagine a collection of "computational particles" sprinkled irregularly on a surface or mixed throughout a volume. These particles are possibly faulty, sensitive to the environment, and may effect actions. In general, the individual particles might be mobile, but the initial programming explorations described in current research papers do not address this possibility.

Each particle has modest computing power and a modest amount of memory. The particles are not synchonized, although it is usually assumed that they compute at similar speeds, since they are all fabricated by the same process. The particles are all programmed identically, although each particle has means for storing local state and for generating random numbers. In general, the particles do not have any a priori knowledge of their positions or orientations.

3.3.2 *Communication model: from wave propagation to pattern formation*

Each particle can communicate with a few nearby neighbors. In amorphous systems of microfabricated components, the particles might communicate via short-distance radio; bioengineered cells might communicate by chemical means. It is assumed that there is some communication radius r, which is large compared with the size of individual particles and small compared with the size of the entire area or volume, and that two particles can communicate if they are within distance r. This crude communication protocol stands from the constraint that communication mechanisms must be independent of the detailed configuration and reliability of the particles. For example, smart paint should be able to determine geometric properties of the surface that it coats without initial knowledge of the positions of the paint's computational particles [1].

Most of the ideas exploited in the field of amorphous computing start from the study of wave propagation: an initial particle "anchor", chosen by a signal from the environment or by generating a random value, broadcasts a message to each of its neighbors. These propagate the message to their neighbors, and so on, to create a diffusion wave that spreads throughout the system. The message can contain a hop count, which each particle can store and increment before rebroadcasting, ignoring any subsequent higher values to prevent the wave from propagating backwards. The hop counts provide estimates of the distance from the anchor: a point reached in n steps will be roughly distance nr away. The quality of this estimate depends on the distribution of the particles. Such relations have been studied in investigation of packet-radio networks [29].

From particles on a surface, one can produce coordinate systems by propagating waves from two anchors. Using three anchors establishes a triangular coordinate system, which can provide better accuracy, especially when augmented by smoohting techniques as discussed by Coore [9] and Nagpal [35]. Wave propagation with hop counts is evocative of the gradients formed by chemical diffusion that are believed to play a role in biological pattern formation. Consequently one can attempt to organize amorphous processes by mimicking gradient phenomena observed in biology. As an example, one can use diffusion waves to produce regions of controlled size, simply by having the processors relay the message only if the hop count is below a designated bound. Once a region is generated in this way, one can use it to control the growth of other regions: this feature has been developed by Nagpal [36] to investigate on how to have programmable materials construct global shapes using biologically-inspired local interactions and origami mathematics.

These diffusion wave mechanisms are well matched to amorphous medium because the gross phenomena of growth, inhibition, and tropism are insensitive to the precise arrangement of the individual particles, as long as the distribution is reasonably dense. In addition, if individual particles do not function, or stop braodcasting, the result will not change very much, so long as there are sufficiently many active particles.

3.3.3 Paradigms for structure

One of the most intersting approach to programming an amorphous computer is the work done by Coore and coworkers [10] on spontaneaously organizing an unstructured collection of processing elements into cooperative groups and hierarchies. This work introduces a structure called an AC Hierarchy, to logically organize processors into groups at different levels of granularity. This is reminiscent of the works on MAS coalition formation. Yet here the idea is to exploit the AC Hierarchy as an active abstraction to build efficient and robust algorithms and simplifies the analysis of their performance.

Each group can operate as a single entity whereas the group members collaborate on specific tasks. The AC hierarchy is then used to provide bounds on the communication latency within a group, for efficient member collaboration, bounds on the size of a group and the proximity of logically close groups. These properties make the AC hierarchy a suitable programming abstraction for implementing a variety of applications, like naming and routing, factoring and mergesort, and distributed sensory control. It simplifies programming with high-level abstractions for partitioning a problem into tasks and multiple levels of tasks, while hiding the details of how a group accomplishes the task. The hierarchies can be used to design efficient resource allocation and to specialize regions within the amorphous computer for different computational or sensory tasks. Elements can be aggregated to increase computational power or increase robustness, which is particularly important in such an unreliable environment. The AC hierarchy bounds provide important timing and locality guarantees which simplify the design and analysis of algorithms using the hierarchy.

These are a few examples of the research done in this new, dynamic domain. The various metaphors used as primitives and organizational principles to try and build effective control of amorphous computing systems sweep a large scope of domains: basic morphogenetic process and their mechanical models of cells by Odell and coworkers [38] serve as source of inspiration for tentative extensions of the message propagation models described above to incorporate local sensing and activation; Physics and conservative processes may provide a framework to understand the collective dynamics of processes exchanging tokens, see Rauch [44] for instance. Such a vivacity of ideas and metaphors is reminiscent of the early days of reactive MAS systems, before the domain eventually somehow settled in dogmatic paradigms.

3.4 Merging both domains

The reason why these two domains seem promising is that an ideal combination of their respective research perspective (incidentally such a convergence gives groundings to Pervasive Intelligence as a field of research) gives insight on the following question, which is central to PI autonomous and adaptive systems: "How to obtain and reproduce coherent behaviors from the cooperation of numerous heterogeneous, situated agents, interacting in unpredictable, irregular and time-varying ways?" One such example of this trend is the SmartIts project [48], which proposes such combinations in the case of patient monitoring: each sensor worn by a patient is capable of adapting to the presence or absence of other sensors, interacting with them and self-organizing so as to give a global picture, emerging from their collective interactions. The collective system hence built may dynamically reconfigure itself as the context (house, hospital, etc.) in which the patient is and the information sources vary.

3.5 From validation process to adaptativeness

The complexity of PI systems will obvisouly not be dealt with conventional validation techniques. The main reasons are the impossibility to (1) manually produce tests or scenarios with sufficient coverage, which is especially the case in rapidly varying environments; (2) formally specify behaviors which will result from probabilistic interactions in unpredictable environments, among different softwares.

As a result the acute question is how to assess the validity of such systems? This question is particularly topical to DAI scientists who will have to work on both individual and collective sides of the artefact behaviors. The only way to answer this question seems to develop systems that would exhibit online adaptativeness rather than a priori proof validity. The capacity to adapt and learn behaviors will undoubtedly be a key characteristic of such agents. This means to work on learning methods which cope with changing environment and imperfect knowledge, which is already the case, for instance in collective robotics [18]. In the long run still, these methods will have to reach industrial standards: e.g. satisfy modularity constraints (incremental design, behavior switches, etc.). Conventional learning techniques poorly meet these constraints [19]. Some recent works, such as those undertaken around ethogenetics [31] may yet provide interesting ways of dealing with such issues. Nonetheless, a huge research domain lay open, which would give rise to new validation techniques and software engineering methods adapted to complex systems, beyond the sole MAS domain.

4. CONCLUSION

In this paper we intended to show that computer science is experiencing a third step in its evolution, with the advent of large distributed systems, organized from the bottom-up, possibly incidental, combination of computational, physical objects in a broad sense. We have described the challenges that this evolution represents and shown the reasons which call for a change in conventional models of design, validation and deployment currently used in software engineering. Beyond this, our objective was to show that DAI, for the very reason that it relies on a distributive stance, owns the promises of possible computational models for such new systems, if only some orthodox traditions were abandonned and other sources of inspiration were embraced. We have concluded by pointing out that so far only the domains of reactive MAS and amorphous computing seemed capable of proposing satisfying solutions. We deliberately did not talk about the ethical and social issues that such computing system evolution certainly lead to. Yet there is little doubt that the adoption or reject of such an evolution will depend on the possibility to take into account in the early stages of design the impact that artefacts and softwares might have on the social environment which they will integrate. Will they lead to new ways of living? of working? Will we be able to accept to depend on systems, autonomous to a great extent? How will we learn to operate with these pervasive distributed, and complex systems? Some of these questions, which will come up sooner or later to DAI scientists, are already tackled by another research domain, that of social computing [7]. There is little doubt still that those questions will have to be at the heart of our future reflexion, as they already are at the center of projects that deal with the social immersion of artificial agents [17] [11].

To conclude with, we hope that this paper enabled the readers to realize the challenge that Pervasive Intelligence represents especially for DAI. A major part of the development of computer science in the 21st century will be centered on this concept, and it will be a shame that DAI which did so much to promote the decentralized mindset in AI would not follow up this evolution which looks massively decentralized.

5. REFERENCES

[1] H. Abelson. Notes on amorphous computing. MIT Department of Electrical Engineering and Computer Science.

[2] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. Knight, R. Nagpal, E. Rauch, G. Sussman, and R. Weiss. Amorphous computing. Technical report, MIT Department of Electrical Engineering and Computer Science, 1999.

[3] D. Abowd. Classroom 2000: An experiment with the instruction of a living educational environment. *IBM Syst. J.*, 1999.

[4] D. Abowd, G. Atkeson, J. Brotherton, T. Enqvist, P. Gulley, and J. Lemon. Investigating the capture, integration and access problem of ubiquitous computing in an educational setting. In M. A. et al, editor, *ACM Conference on Human Factors in Computing Systems*, pages 440–447. ACM Press/ Addison Wesley, 1998.

[5] Amorphous computing homepage, 2001. http://www.swiss.ai.mit.edu/ projects/ amorphous/.

[6] Bluetooth technology homepage, 2001. http://www.bluethooth.com/ technology/.

[7] R. Braham and R. Comerford. Sharing virtual worlds: Avatars, agents, and social computing. *IEEE Spectrum*, 1997.

[8] Special issue on embedding the internet. ACM Press, 2000.

[9] D. Coore. Establishing a coordinate system on an amorphous computer. In *1998 MIT Student Workshop on High Performance Computing in Science and Engineering*, 1998.

[10] D. Coore, R. Nagpal, and R. Weiss. Paradigms for structure in an amorphous computer. Technical report, MIT Department of Electrical Engineering and Computer Science, 1997.

[11] K. Dautenham. Embodiment and interaction in socially intelligent life-like agents. In *Computation for Metaphors, Analogy and Agent*, 1999.

[12] Y. Demazeau, editor. *Proceedings of MAAMAW 2001.* LNCS Series Springer-Verlag, 1998.

[13] E. D'Hondt. Exploring the amorphous computing paradigm. Technical report, VUB, Bruxelles, 2000.

[14] A. Drogoul. When ants play chess (or can strategies emerge from tactical behaviors?). In *From reaction to cognition*, 1995.

[15] A. Drogoul. Systèmes multi-agents. Dossier d'habilitation à diriger des recherches, 2000.

[16] A. Drogoul and J. Meyer, editors. *Intelligence Artificielle Située.* Hermès, 1999.

[17] A. Drogoul and S. Picault. Microbes: vers des collectivités de robots socialement situés. In Gleizes and Marcenac, editors, *Actes des JFIADSMA'99*, pages 265–278. Hermès, 1999.

[18] A. Drogoul, M. Tambe, and T. Fukuda, editors. *Collective Robotics.* LNAI Series Springer-Verlag, 1998.

[19] A. Drogoul and J.-D. Zucker. Methodological issues for designing multi-agent systems with machine learning techniques: Capitalizing experiences from the robocup challenge. Technical report, Université de Paris 6, 1998.

[20] E. Durfee, S. Kraus, H. Natashima, and M. Tamber, editors. *ICMAS 2000.* IEEE Editions, 2000.

[21] E. Ertico, editor. *Towards an intelligent road system.*

Artech House, 1995.

[22] T. Finin. Kqml as an agent communication language. In *Proc. of the 3rd Int. Conf. on Information and Knowledge Management*. ACM Press, 1994.

[23] Foundation for intelligent physical agents homepage, 2001. http://www.fipa.org.

[24] S. Huang and W. Ren. Autonomous intelligent vehicle and its performance in automated traffic systems. *International Journal of Control*, 1999.

[25] 802.11 protocol homepage, 2001. http://www.manta.ieee.org/ groups/ 802/ 11/.

[26] Information page, 2001. http://www.ipv6.org/.

[27] N. Jennings. On agent-based software engineering. *AI Journal*, 2000.

[28] N. Jennings and K. Sycara, editors. *Autonomous Agents and Multi-Agent Systems*. Kluwer, 2001.

[29] L. Kleinrock and J. Silvester. Optimum transmission radii in packet radio networks or why six is a magic number. In *National Telecommunications Conference, Birmimgham, Al*, 1978.

[30] T. Knight and G. Sussman. Cellular gate technology. In *First International Conference on Unconventional Models of Computation, Auckland NZ*, 1998.

[31] S. Landau and S. Picault. Developing agents populations with ethogenetics. In *Proc. of WRAC01 workshop*, 2001.

[32] J. Licklider. Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics*, 1960.

[33] Overview, 2001. http://www.microopticalcorp.com/.

[34] C. Mozer. The neural network house: An environment that adapts to its inhabitants. In *Proceedings of the AAAI Spring Symposium on Intelligent Environments*, 1998.

[35] R. Nagpal. Organizing a global coordinate system from local information on an amorphous computer. Technical report, MIT Department of Electrical Engineering and Computer Science, 1999.

[36] R. Nagpal. *Programmable self-assembly: constructing global shape using biologically-inspired local interactions and origami mathematics*. PhD thesis, MIT Department of Electrical Engineering and Computer Science, 2001.

[37] Pervasive computing 2001, 2001. http://www.nist.gov/pc2001/.

[38] G. Odell, G. Oster, P. Alberch, and B. Burnside. The mechanical basis of morphogenesis, 1. epithelial folding and invagination. *Developmental Biology*, 1981.

[39] H. Parunak. Go to the ant: Engineering principles from natural agent systems. *Annals of Operations Research 75*, 1997.

[40] J. Pascoe, , S. Ryan, and R. Morse. Issues in developing context-aware computing. In *Proc. Intern. Symposium on Handheld and Ubiquitous Computing*, 1999.

[41] A. Pentland. Augmented reality throuh wearable computing. *Presence: Teleoper. Virtual Environ.*, 1997.

[42] Philips laboratories homepage, 2001. http://www.research.philips.com/ generalinfo/ special/ ambintel/.

[43] A. Rao and M. Georgeff. Bdi agents from theory to practice. Technical report, Technical Note 56, AAII, 1995.

[44] E. Rauch. Discrete, amorphous physical models. Master's thesis, MIT Department of Electrical Engineering and Computer Science, 1999.

[45] J. Rekimoto, Y. Ayatsuka, and K. Hayashi. Augment-able reality: Situated communication through physical and digital spaces. In *Proc. ISWC'98*, 1998.

[46] J. Rekimoto and K. Nagao. The world through the computer: computer augmented interaction with real world environments. In *Proceedings of the 8th ACM Symposium on User Interface and Software Technology*, 1995.

[47] M. Resnick. Beyond the centralized mindset. In *WOFCAI'92*, pages 369–396, 1992.

[48] Smartits homepage, 2001. http://www.vision.ethz.ch/ projects/ smartits/ smartits.html.

[49] Jini technology homepage, 1999. http://www.sun.com/jini.

[50] S. Thrun, M. Bennewitz, W. Burgard, B. Cremers, F. Dellaert, D. Fox, D. Hûhnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schultz. Minerva : A second-generation museum tour-guide robot. In *Proceedings of ICRA'99*, 1999.

[51] M. Turk, editor. *Proceedings of the Perceptual User Interfaces Workshop*, 1997.

[52] M. Turk, editor. *Proceedings of the Perceptual User Interfaces Workshop*, 1998.

[53] Links, 2001. http:/ /homepage1.nifty.com/ konomi/ shinichi/ ubicomp.html.

[54] M. Weiser. The computer for the twenty-first century. *Scientific American*, 1991.

[55] M. Weiser. Some computer science issues in ubiquitous computing. *Commun. ACM*, 1993.

[56] W. Wolf. Hardware-software co-design of embedded systems. *IEEE SMC*, 1994.