Distributed Computing on the Move: From mobile computing to cooperative robotics and nanorobotics

[Position Paper]

Xavier Défago
School of Knowledge Science
Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Tatsunokuchi, Nomigun, Ishikawa 923-1292, Japan
defago@jaist.ac.jp

ABSTRACT

Distributed computing is beginning to extend its scope to address problems relevant to a mobile environment (mobile computing). For the most part, current research efforts in mobile computing and ad hoc networking are implicitly aimed at mobile telephony and the emerging field of ubiquitous computing.

More generally, this paper considers distributed computing environments in which hosts are (physically) mobile. The paper essentially advocates the integration into the picture of other important challenges that seem to represent the next logical step beyond mobile computing, namely cooperative robotics and nanorobotics.

Conceptually, the main difference introduced by mobile computing over classical distributed computing is the fact that the physical position of hosts becomes an important parameter which can no longer be abstracted out. Cooperative robotics brings the issue further by allowing (and requiring) to actually *control* the physical position of the hosts.

Categories and Subject Descriptors

A.1 [**General Literature**]: Introductory and Survey; E.4 [**Data**]: Coding and Information Theory—*Formal models of communication*; F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*relation between models, self-modifying machines, unbounded-action devices (network of machines)*; I.6.5 [**Computing Methodologies**]: Simulation and Modeling—*model development*

General Terms

Algorithms, design, theory

Keywords

self-stabilization, mobility, cooperative robotics, nanorobotics

POMC '01 Newport, Rhode Island USA

1. INTRODUCTION

The field of distributed computing originates from parallel computing and belongs to the study of concurrent systems. As the shared memory models (e.g., PRAM and its siblings) turned out to be inadequate for distributed memory machines, it became necessary to define new models. This shift did not only trigger changes in the system model, but also led to the definition of new problems (e.g., Causal Broadcast).

From its origin, distributed computing gradually evolved in answer to new environments and requirements (e.g., distributed databases, reservation systems, Internet applications), and developed into a well established domain that contains both practical applications and a strong theoretical foundation. This evolution has broaden the field to include aspects with little relevance to the original concern (i.e., numerical computations), such as reliability, security, or scalability.

Current developments in mobile networking, such as ad hoc networks, provide an interesting challenge for research in distributed computing. Indeed, in mobile systems, the topology of the connectivity graph changes over time. But, and this is more important, the *definition* of important problems depends explicitly on the physical position of hosts [17, 16, 27]. As a simple example one can consider a mobile phone application which would allow a user to establish a communication with the nearest available taxi. Obviously, the physical position of the hosts can no longer be abstracted out entirely and must be somehow integrated into the model. In fact, this is probably what constitutes the most fundamental difference introduced by mobile computing over classical distributed computing.[1]

So far, a large majority of developments in mobile computing have targeted application in mobile telecommunications. More recently, ubiquitous computing [34] has also started to draw a certain attention (e.g., disappearing computer initiative [31], terminodes [14]). These two fields are in fact closely related and complementary. The former aims at

[1]In this paper, I use the terms "mobile" and "mobility" to refer to the *physical* mobility of hosts. Thus, mobile computing, as I understand it, does not include the topic of mobile agents since mobility, in the latter case, is just a metaphor. Of course, this does not preclude the application of *solutions* based on mobile agents to solve *problems* in mobile computing.

developing an infrastructure, whereas the latter is more directly focused on applications. Anyway, these developments gradually find an echo in the distributed systems community, as testified by the increasing number of system models and abstract problems that are defined in this context [13, 23, 32].

This paper describes two other application fields that lay beyond mobile computing, namely cooperative robotics and cooperative nanorobotics, and provides a short survey. These two fields seem to represent the next logical steps after mobile/ubiquitous computing. Briefly, cooperative robotics is concerned with the coordinated control of teams of robots. Similarly, cooperative nanorobotics is a problem applied to nanorobots (molecular level robots), with a difference in scale (small size, large numbers) and capabilities (self-replication). The paper advocates the development of a computation theoretic foundation to those problems, using the same approach that is prevalent in concurrent and distributed systems [21]. This typically includes the definition of proper system models, the rigorous specification of abstract problems (e.g., Leader Election, Consensus, Byzantine Agreement in distributed computing), the study of the relation between problems, as well as the design, verification, and analysis of algorithms to solve those problems.

The rest of this paper is structured as follows. Section 2 presents the context by explaining what are cooperative robotics and cooperative nanorobotics. Section 3 describes cooperative computing as the blending of distributed/mobile computing and cooperative robotics/nanorobotics. Reaching this goal will take three phases: applications (Sect. 4), problems and models (ref. 5), and algorithms and verification (Sect. 6). Section 7 concludes the paper and tries to open the discussion.

2. CONTEXT

This section briefly describes the two fields of cooperative robotics and cooperative nanorobotics. Although considering these two fields separately might sound a little peculiar at first, this section hopefully makes it clear that there is a fundamental difference between them, namely the potential ability of nanorobots to self-replicate.

2.1 Cooperative robotics

As the production cost of robots decreases, it is more and more tempting to consider applications involving teams of robots rather than a single entity. This motivation is even greater for small robots such as micro electro-mechanical systems (MEMS).

However, turning a collection of robots into a single coherent team is definitely not a trivial problem. This is nevertheless the challenge addressed by the field of cooperative robotics. In fact, cooperative robotics can hardly be considered as a coherent mono-cultural research field. It is rather an intensely multi-disciplinary field with contributions coming from fields ranging from electronics to artificial intelligence, as well as from biology to economics.

Since a robot can be considered as a "computer with legs," it is natural to consider a team of robots as a kind of mobile distributed system. There are however two fundamental differences with conventional distributed systems. Firstly, robots usually require and use knowledge about their physical location (just like mobile computing). Secondly, and this is more important, robots must *control* their own motion.

2.2 Cooperative nanorobotics

The field of cooperative nanorobotics also considers the coordination of teams of robots, and hence faces the same difficulties. However, nanorobotics considers robots that are built at a molecular-level [8, 26]. This means that individual parts of the robot, such as engine rotors, bearings, pumps are each made of single molecules, thus resulting in extremely small robots. According to researchers in molecular nanotechnology, although no such robot has presently been synthesized, it would theoretically be possible to build robots barely larger than most viruses (e.g., *Influenza virus*, about 100nm) but smaller than a bacterium (e.g., *Escherichia Coli*, about 2μm), let alone animal cells (about 50μm).

The extremely small size of nanorobots opens up the horizon to a vast number of potential applications (see Sect. 4). The bottom line is that a single nanorobot is unlikely to have any noticeable effect at the human-scale. So, almost every applications of nanorobotics imagined so far involve large teams of robots rather than a single entity. Although the problem of coordinating such teams is similar to that of cooperative robotics, it is occurs at a much greater scale. Indeed, assuming a volume of one liter with a concentration of 10^{-3} of 1μm nanorobots, this volume would contain in the order of 10^{12} such robots. These figures show that scalability issues in such a system are of vital importance.

If it were only for the difference in scale, cooperative nanorobotics would essential be the same as cooperative robotics. There is however a fundamental difference between the two contexts beyond the scale factor. It is the potential ability of nanorobots to self-replicate [24], i.e., to make a *physical* copy of themselves. This should of course not be mistaken for process or data replication.

3. COOPERATIVE COMPUTING

As mentioned in the introduction, this paper advocates the development of a sound theoretical foundation to address the problems of cooperative robotics and nanorobotics. Even though there is little hope that any application involving nanorobotics will be realized in the foreseeable future, there are several reasons why the problem of coordination should be considered at such an early stage:

- *Accelerating development.*
  Without a good view of the objectives, developments will continue to follow a bottom-up approach and are likely to go into a multitude of directions. This "brute-force approach" is fine as long as the problem is small or the available resources are infinite. Hence, the investigation can be complemented by a top-down approach in order to provide a goal and reduce the number of tracks to investigate.

- *Orienting technological developments.*
  Studying the minimum conditions under which particular problems are solvable is important to understand what must be developed in order to implement some specific applications.[2]

- *Avoiding pitfalls and dead ends.*
  Isolating the problems that are impossible, undecid-

[2]Note that the argument can also be reversed in the sense that it would allow to avoid developing too much.

able, or intractable will help to avoid developments in directions that are likely to lead to a dead end.

- *Validating existing and future solutions*
  It is always easy to provide a solution to a vaguely defined problem, and then argue that "it should work in most cases!" However, engineering complex systems usually requires more than a vague intuition, especially if the system needs to be trusted afterwards. The existence of a rigorous specification of the problems is a necessary condition to prove the correctness of proposed solutions.

So far, the problem of coordinating a team of robots has been addressed with many different approaches. However, there is still hardly any research effort that aims at developing a formal framework to study the problem, such as what exists in the context of concurrent and distributed systems. Again, the message that this paper tries to convey is that distributed computing can also serve as a starting point to define the framework of "cooperative computing," namely, a theoretical framework for cooperative robotics, based on the formal specification of problems. The development of such a framework will take a top-down approach and involve three phases (not purely sequential):

Phase 1: *Applications.* (Sect. 4)
The first phase will necessarily be application-oriented because it will define what objectives to reach. It will first consist in defining various applications of cooperative robotics (resp. nanorobotics). These applications will motivate the specification of specific problems, and orient the definition of system models.

Phase 2: *Generic problems and models.* (Sect. 5)
The second phase will aim at defining a generic framework by isolating recurrent problems, and identifying the parameters and their effect in the system models. This step will also investigate the mutual relationship between problems and models, such as equivalence between problems, etc.

Phase 3: *Algorithms and verification.* (Sect. 6))
The third phase will be concerned with the solutions to the problems. For instance, developing (and proving) algorithms to solve some problem $\mathfrak{P}$, finding the minimal requirements under which some other problem $\mathfrak{P}'$ is solvable, etc.

All of these phases can draw from previous experience and development in other fields. The following three sections (Sect. 4–6) present results and developments that are relevant to each phase.

4. PHASE 1: APPLICATIONS

The first phase defines the objectives. This can only be done by identifying as many applications that would use the technology, and anticipating the difficulties. The goal is of course not to implement those applications (at least for a while), but rather to understand better their respective requirements. This section gives a short survey of applications related to cooperative nanorobotics that have been proposed in the literature.

A detailed specification of each application and its requirements will generate a collection of application-specific problems and models.

4.1 Exploration

The unmanned exploration of unaccessible environments (e.g., space, deep sea) is an obvious application for both cooperative robotics and nanorobotics.

At the macro level, several ongoing research or development projects testify of this interest. For instance, the NASA Jet Propulsion Laboratory is developing an excavation system to (literally) prepare the ground for manned expeditions to Mars. This system will consist of several hundreds of small semi-autonomous rovers controlled by a base station [29].

Ferris [9] proposes a more futuristic view of space exploration in which self-replicating groups of nanorobots colonize the galaxy by building small base stations on planets and asteroids, thus providing the basic infrastructure for an interstellar information network. Even though this might sound a little far-fetched by current standards, it might at least inspire future programs on the exploration of our solar system.

4.2 Manufacturing

Many texts on nanotechnology, including Feynman's seminal talk [10], present molecular manufacturing as one of the main applications of the technology [8, 26]. This is partly due to the sense of wonder that manipulating individual atoms inspires, but also because this is an important milestone towards achieving self-replication.

The manufacturing of human-scale objects by nanofactories (also called assemblers) is unlikely to be achieved by a single assembler. Instead, this will almost certainly necessitate very large teams of assemblers to cooperate. Merkle [25] proposes a design for building such an assembler, and acknowledges the necessity of forming teams. However, he proposes a centralized architecture combined with a model analogous to the single instruction multiple data model (SIMD) of parallel computation; a model that is well-know for its limitations with respect to practical scalability.

4.3 Medicine and surgery

There are numerous medical applications using colonies of MEMS or nanorobots, such as micro-surgery. Some medical applications are even already under development.

For instance, researchers are developing nanorobots (sometimes called NEMS[3]) made of a combination of organic and inorganic components [3, 30]. One of their goals is to enable more efficient medicine practices by making it possible to control the precise location and the exact amount of drugs delivered in the body. Rubinstein [28] also drafts the picture of a simple robot that could be built with our current technology. However, the communication and cooperation between nanorobots remains largely ignored.

Freitas Jr. [11] proposes many medical applications, among which the ability of nanorobots to perform surgery at the molecular level. He also states that "nanodevices must be made to communicate with each other in order to: (1) coordinate complex, large-scale cooperative activities, (2) pass along relevant sensory, messaging, navigational, and other operational data, and (3) monitor collective task progress," and describes in details how nanorobots could communicate in the context of the human body.

4.4 Smart materials

[3]NEMS: nano electro-mechanical systems.

Other applications include "smart materials"; materials made of a multitude of tiny modules that link to each other forming some large structure. Such a system could then change its shape and physical properties (e.g., from solid to liquid and back).

5. PHASE 2: PROBLEMS AND MODELS

The second phase aims at genericity through a careful analysis of the application-specific problems and models devised during the first phase. The main objective is to identify and isolate recurrent problems and define generic system models.

5.1 Problems

One of the greatest strength of distributed computing over less formalized fields (e.g., cooperative robotics) is the fact that several recurrent problems have been isolated, such as distributed mutual exclusion, consensus, leader election, distributed commit [2, 22]. These problems have been identified as central problems to solve *classes* of applications rather than specific ones.

The study of recurrent problems in the context of both mobile computing and cooperative computing involves two complementary paths. Firstly, problems that are already defined in the context of distributed computing should be adapted to the new domain (e.g., mobile computing or cooperative computing), whenever applicable. Secondly, problems that are specific to the characteristics of the new domain should be identified (e.g., location-dependent problems for mobile computing).

In mobile computing, the formal study of generic problems has begun only recently. For instance, Malpani et al. [23] propose an algorithm for Leader Election in mobile networks. As for location-dependent specifications, Ko and Vaidya [17] discuss the problem of geocasting, which consists in multicasting information to all hosts that are located within a certain geographical perimeter at a given time.

At the time of writing and to the best of my knowledge, there is only one formal study of a recurrent problem related to cooperative computing. The most closely related work is due to Walter et al. [33]. The context is metamorphic robots, namely robots that can modify their own shape to adapt to new environmental contexts. In the paper, they take a rigorous approach to solve the problem of reconfiguration. Their work is of course certainly relevant to cooperative computing and it could also probably address the question of coordinated motion (teams moving in unison).

5.2 Models

With cooperative robotics, the system has more degrees of liberty than conventional distributed systems. Hence, one has to consider the following aspects of a model as being orthogonal: *communication model* and *system model*. The former defines how entities can interact, whereas the latter defines how they move.

5.2.1 Communication Models

In order to cooperate, robots must be able to communicate in a way or another. The following enumeration of communication models is an extension to the one given by Prakash and Baldoni [27].

No communication. The mobile hosts do not communicate directly. Instead, they act according to their own perception of the environment. Assuming that they also have the means to act on that environment, the hosts can still interact, albeit only indirectly. This can be understood as a limited form of communication. For instance, consider a system in which a robot enters a room and locks the only entrance door. Other robots are hence unable to enter that room anymore because the door is locked. One can consider that the state of the door (locked) is like a boolean message sent by the first robot to the others. In this case, this is a physical form of mutual exclusion.

Cellular network model. The area is divided into cells, each of which is managed by a base station. A backbone network interconnects the base stations. Two mobile hosts cannot however establish a direct connection. They must instead rely on a base station in order to communicate.

In traditional settings (e.g., mobile phone systems), the base stations are fixed. In a more flexible variant, called the virtual cellular network model, base stations are also mobile and the backbone is built using wireless communication.

Ad hoc network model. Mobile hosts use wireless communication, but do not rely on any specific infrastructure (i.e., base stations network). Instead, each mobile host can route messages on behalf of others. This makes it possible for two mobile hosts to communicate even without a direct line-of-sight, as long as other hosts can relay the communication and thus create a path between them.

Alternate network models. Depending on the environment, some communication approaches might be out of the question. For instance, deep space exploration projects are obviously bound to fail if they rely on ultrasonic communication. In some applications, none of the existing techniques might be acceptable. For instance, in nanomedicine applications, conventional wireless communication might prove lethal, whereas living cells naturally use a whole set of communication tools already (e.g., hormones, neurotransmitters) [1] that might be used or mimicked.

These considerations are important, because the communication medium cannot always be abstracted out completely in the network model. Some communication media allow for interactions that are impossible with others, thus having a direct impact on the way to model communication links. For instance, broadcast media behave differently from point-to-point media with respect to contention.

5.2.2 System Models

This section presents the different system models relevant to cooperative robotics. Some models have been developed in ad hoc networking and metamorphic robotics that could certainly provide a good starting point for a proper model definition for cooperative robotics.

Ad hoc networking. Several models, problems, algorithms are currently being defined in the context of ad hoc networking. For instance, Prakash and Baldoni [27] present a model in which the position of hosts is explicit. They propose an architecture for group communication in the context of mobile hosts, in which some subproblem definitions depend explicitly on the location of hosts. Hatzis et al. [13] propose a model for ad hoc networks, identify Leader Election as a fundamental problem in this context, and define the classes of *compulsory* and *non-compulsory* protocols. In short, compulsory protocols determine the motion of the hosts, whereas non-compulsory protocols do not affect it.

Metamorphic robots. There has been only very few attempts at providing a theoretical foundation to the field. As far as I know, the most relevant work is a system model and reconfiguration algorithm for metamorphic robots, proposed by Walter et al. [33]. Even though the context is a little different, their model could provide an adequate starting point for cooperative robotics. Other developments related to this field fall more or less into three categories.

1. Developments specific to a given architecture (see introduction in [33])

2. Ad hoc solutions [15]

3. Heuristic approach [18]

Cooperative robotics and nanorobotics. As mentioned earlier, I know of no formal definition so far of a system model for cooperative robotics. The most relevant works are the model developed by Walter et al. [33] for metamorphic robots, the models and problems developed in the context of mobile networking [17, 23], and the vast legacy of distributed computing [2, 22], concurrency [20], and self-stabilization [7].

The work done on the specification of concurrent systems and on self-stabilization is fairly general. In fact, it might not even be necessary to develop new models for cooperative robotics, and extending existing ones might turn out to be sufficient. However, some additional problems related to controlling or coping with the physical location of hosts will certainly have to be defined.

6. PHASE 3: ALGORITHMS AND VERIFICATION

In spite of the lack of formally defined problems, the coordination of independent entities has been a topic of active researches. Several kinds of solutions have been proposed in partial answer to the (yet unspecified[4]) problem. Even though none of these solution has been proved correct yet, they might bring interesting insight as to how formalized coordination problems could be solved.

6.1 Ad hoc approach

Several practical applications involving teams of robots have actually been built already, or are under development. The vast majority of these systems however take a very pragmatic approach, without any concern for genericity. For instance, the Mars nanorovers project [29] mentioned earlier considers a fully centralized approach in which the landing pod acts as a fixed base station and controls the 700 rovers. In addition to the single point of failure introduced by the base station, there is the obvious question of scalability. Of course, the very tight deadlines under which the JPL must build this project do not allow for a deep exploration of new theoretical concepts. So, these concepts must be developed independently for the benefit of future projects that might involve several orders of magnitude more robots.

[4]Pun obviously intended!

6.2 Bio-inspiration and bio-mimetism

Other attempts at developing self-organizing systems consists in finding examples in nature. For instance, a popular approach is to imitate the organization of social insects, such as bees or ants [18]. Most of the modeling done in this context concerns the simulation of systems based on stochastic processes. Even though some collaboration problems are described (e.g., ant cemetery, stick pulling), I have not yet found a rigorous specification for any of them. The researches done in this direction yield very interesting insight for the solution of problems in which suboptimal solutions are acceptable.

6.3 Optimization approach

One approach to self-organization consists in transforming the problem into an optimization problem, and then apply adequate techniques, such as simplex, genetic algorithms, or economical models [5, 35]. For instance, the main idea in the latter case is to model the system as an economical market in which the efforts of individual components to maximize their personal value is supposed to maximize the global value of the system, thus eventually reaching some objectives. Of course, these approach do not yield optimal results.

6.4 Emergent behavior and artificial life

Some researches study the global behavior that spontaneously emerges from the interaction of simple entities. In the context of social science, researchers are for instance interested in the mechanisms leading to the spontaneous creation of norms in social groups [12]. Similarly, in the context of artificial life, many researchers study the emergence of unanticipated social structures from the interaction of simple entities.

6.5 Self-stabilization

Self-stabilization [7] is concerned with systems which, given any initial configuration configuration, are guaranteed to eventually reach some desirable target configuration. The idea underlying this approach was initially proposed by Dijkstra [6], and later revived by Lamport [19]. Although the field does not provide solutions or algorithms that are directly relevant to cooperative robotics, it provides an important theoretical background to developing (and proving) new algorithms.

7. CONCLUSION

Cooperative robotics still lacks the rigorous theoretical foundation developed in other fields, such as in distributed computing and concurrent systems. Since the field of distributed robotics can be seen as a subclass of concurrent systems, it would be natural to think that an adaptation is straightforward and nothing needs to be developed. Even though this might actually turn out to be true for the definition of models, new location-dependent problems will certainly have to be specified.

In spite of some recent work in a similar direction [33], the field is still largely unexplored and promises for many interesting challenges. Other developments realized so far were mostly based on ad hoc solutions or some heuristics. Although interesting and insightful, using a rigorous approach to define the models and the problems will surely allow for a better understanding of the field, and will reduce the risk of misinterpretation and misunderstandings. This would give a context in which more generic algorithms might be developed and analyzed using properly defined complexity measures rather than pure intuition and subjectivity.

As Crowcroft [4] wrote recently, "many of the most difficult problems will be in areas where we have only scratched the surface—in *novel models for emergent behavior*;[5] ..." But, rather than seeing the problems as difficult, I would rather consider them as challenging and begin scratching deeper and deeper.

8. ACKNOWLEDGMENTS

I would like to thank Akihiko Konagaya and Péter Urbán for their valuable comments on an earlier version of this paper, as well as their encouragements.

9. REFERENCES

[1] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Essential Cell Biology: An introduction to molecular biology of the cell*. Garland, 1998.

[2] H. Attiya and J. L. Welch. *Distributed Computing: Fundamentals, simulations, and advanced topics*. McGraw-Hill Publishing, Oct. 1998.

[3] G. D. Bachand and C. D. Montemagno. Constructing organic/inorganic nems devices powered by biomolecular motors. *Biomedical Microdevices*, 2:179–184, 2000.

[4] J. Crowcroft. Never lost, never forgotten. *Commun. ACM*, 44(3):81, Mar. 2001.

[5] M. B. Dias and A. Stentz. A free market architecture for distributed control of a multirobot system. In *Proc. of the 6th Int'l Conf. on Intelligent Autonomous Systems (IAS)*, Venice, Italy, July 2000.

[6] E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11):643–644, Nov. 1974.

[7] S. Dolev. *Self-Stabilization*. MIT Press, 2000.

[8] K. E. Drexler. *Nanosystems: molecular machinery, manufacturing, and computation*. Wiley Interscience, 1992.

[9] T. Ferris. Interstellar spaceflight: Can we travel to other stars? *Scientific American Quarterly*, pages 88–91, Spring 1999. Special issue (the future of space exploration).

[10] R. P. Feynman. There's plenty of room at the bottom. *Engineering and Science*, Feb. 1960. http://www.zyvex.com/nanotech/feynman.html.

[11] R. A. Freitas Jr. *Nanomedicine: Basic capabilities*, volume 1. Landes bioscience, 1999. http://www.nanomedicine.com/.

[12] T. Hashimoto and S. Egashira. Formation of social norms in communicating agents with cognitive frameworks. *Journal of Systems Science and Complexity*, 14(1):54–74, 2001.

[5]Italics mine.

[13] K. P. Hatzis, G. P. Pentaris, P. G. Spirakis, V. T. Tampakas, and R. B. Tan. Fundamental control algorithms in mobile networks. In *Proc. of the 11th ACM Symp. on Parallel Algorithms and Architectures (SPAA'99)*, pages 251–260, Saint-Malo, France, June 1999.

[14] J.-P. Hubaux, T. Gross, J.-Y. L. Boudec, and M. Vetterli. Towards self-organized mobile ad hoc networks: the terminodes project. *IEEE Communications*, 39(1):118–124, Jan. 2001.

[15] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Mobile networking for Smart Dust. In *ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom 99)*, pages 271–278, Seattle, WA, Aug. 1999.

[16] Y. Ko and N. H. Vaidya. Location-aided routing (lar) mobile ad hoc networks. In *ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom 98)*, pages 66–75, Dallas, TX, USA, Oct. 1998.

[17] Y. Ko and N. H. Vaidya. Geocasting in mobile ad hoc networks: Location-based multicast algorithms. In *Proc. of the 2nd IEEE Workshop on Mobile Computer Systems and Applications (WMCSA'99)*, New Orleans, LA, USA, Feb. 1999.

[18] M. J. B. Krieger, J.-B. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799):992–995, Aug. 2000.

[19] L. Lamport. Solved problems, unsolved problems and non-problems in concurrency. In *Proc. of the 3rd Annual ACM Symp. on Principles of Distributed Computing (PODC-3)*, pages 63–67, Aug. 1984. invited address.

[20] L. Lamport. Distributed operating systems: Theory and practice. In Paker, Banâtre, and Bozyigit, editors, *A Formal Basis for the Specification of Concurrent Systems*, pages 1–46. Springer-Verlag, 1987.

[21] L. Lamport. A simple approach to specifying concurrent systems. *Commun. ACM*, 32(1):32–45, Jan. 1989.

[22] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[23] N. Malpani, J. L. Welch, and N. H. Vaidya. Leader Election algorithms for mobile ad hoc networks. In *Proc. of the 4th Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Boston, USA, Aug. 2000.

[24] R. C. Merkle. Self replicating systems and molecular manufacturing. *Journal of the British Interplanetary Society*, 45:407–413, 1992. http://www.zyvex.com/nanotech/selfRepJBIS.html.

[25] R. C. Merkle. A simple assembler. In *NASA Ames Computa-tional Molecular Nanotechnology Workshop*, Mar. 1996. http://www.zyvex.com/nanotech/nano4/merklePaper.html.

[26] C. Peterson. Taking technology to the molecular level. *IEEE Computer*, 33(1):46–53, Jan. 2000.

[27] R. Prakash and R. Baldoni. Architecture for group communication in mobile systems. In *Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems (SRDS-17)*, pages 235–242, West Lafayette, IN, USA, Oct. 1998.

[28] L. Rubinstein. A practical nanorobot for treatment of various medical problems. In *Proc. of the 8th Foresight*

Conf. on Molecular Nanotechnology., Nov. 2000.

[29] K. Schreiner. NASA's JPL nanorover outposts project develops colony of solar-powered nanorovers. *IEEE DS Online*, 2(3), 2001.

[30] R. K. Soong, G. D. Bachand, H. P. Neves, A. G. Olkhovets, H. G. Craighead, and C. D. Montemagno. Powering an inorganic nanodevice with a biomolecular motor. *Science*, 290(5496):1555–1558, Nov. 2000.

[31] L. Streitz, N. Goulden, S. Lalis, and P. Nixon. The disappearing computer initiative. Project home page, 2001. http://www.disappearing-computer.net.

[32] J. Walter, J. L. Welch, and N. H. Vaidya. A mutual exclusion algorithm for ad hoc mobile networks. In *1998 Dial M for Mobility workshop*, Dallas, TX, USA, Oct. 1998.

[33] J. E. Walter, J. L. Welch, and N. M. Amato. Distributed reconfiguration of hexagonal metamorphic robots. *IEEE Trans. on Robotics and Automation*, 2001. *To appear in special issue on multi-robots systems.*

[34] M. Weiser. The computer for the twenty first century. *Scientific American*, pages 94–101, Sept. 1991.

[35] D. Wolpert, K. Wheeler, and K. Tumer. Collective intelligence for control of distributed dynamical systems. *Europhysics Letters*, 49(6):708–714, Mar. 2000.