Biology-Inspired Approaches to Peer-to-Peer Computing in BISON *

Alberto Montresor and Ozalp Babaoglu

Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7, 40127 Bologna, (Italy), E-mail: {montresor,babaoglu}@CS.UniBO.IT
Tel. +39 051 2094973 Fax +39 051 2094510

Summary. BISON is a research project funded by the European Commission that is developing new techniques and paradigms for the construction of robust, self-organizing and self-repairing information systems as ensembles of autonomous agents that mimic the behavior of some natural or biological process. In this paper we give a brief overview of BISON, discuss some preliminary results for peer-to-peer systems, describe the ongoing work and indicate future research directions that appear to be promising.

1 Introduction

Modern information systems that are distributed are gaining an increasing importance in our every day lives. As access to networked applications become omnipresent through PC's, hand-held and wireless devices, more and more economical, social and cultural transactions are becoming dependent on the reliability and availability of distributed services.

As a consequence of the increasing demands placed by users upon networked environments, the complexity of modern distributed systems has reached a level that puts them beyond our ability to design, deploy, manage and keep functioning correctly through traditional techniques. Part of the problem is due to the sheer size that these systems may reach, with millions of users and interconnected devices. The other aspect of the problem is due to the extremely complex interactions that may result among components even when their numbers are modest. Our current understanding of these systems is such that even minor perturbations (e.g., a software upgrade, a failure) in some remote corner of the system will often have unforeseen, and at times catastrophic, global repercussions. In addition to being fragile, many situations (e.g., adding/removing components) arising from their highly dynamic

* This work was partially supported by the Future & Emerging Technologies unit of the European Commission through Project BISON (IST-2001-38923).

environments require manual intervention to keep information systems functioning.

In order to deal with the scale and dynamism that characterize modern distributed systems, we believe that a paradigm shift is required that includes self-organization, adaptation and resilience as intrinsic properties rather than as afterthought. With this motivation, we have initiated BISON (*Biology-Inspired techniques for Self Organization in dynamic Networks*), an international research project funded by the European Commission. BISON draws inspiration from natural and biological processes to devise appropriate techniques and tools to achieve the proposed paradigm shift and to enable the construction of robust and self-organizing distributed systems for deployment in highly dynamic modern network environments.

2 Overview of the BISON Project

Nature and biology have been a rich source of inspiration for computer scientists. *Genetic algorithms* [4], *neural networks* [8] and *simulated annealing* [10] are examples where different natural processes have been mimicked algorithmically to successfully solve important practical problems. More recently, *social insects* [2] have been added to this list, inspiring biomimetic Algorithms to solve combinatorial optimization problems. *Immune networks* [11] represent a recent frontier in this area.

These processes are examples of *complex adaptive systems (CAS)* [7] that arise in a variety of biological, social and economical phenomena. In the CAS framework, a system consists of large numbers of *autonomous agents* that individually have very simple behavior and that interact with each other in very simple ways. CAS are characterized by total lack of centralized coordination. Despite the simplicity of their components, CAS typically exhibit what is called *emergent behavior* that is surprisingly complex and unpredictable [7]. Furthermore, the collective behavior of a well-tuned CAS is highly adaptive to changing environmental conditions or unforeseen scenarios, is robust to deviant behavior (modeling failures) and is self-organizing towards desirable configurations.

Parallels between CAS and advanced information systems are immediate. BISON will exploit this fact to explore techniques and ideas derived from CAS in order to derive robust, self-organizing and self-repairing solutions to problems arising in dynamic networks through ensembles of autonomous agents that mimic the behavior of some natural or biological process. In our opinion, exploitation of CAS techniques will enable developers to meet the challenges arising in dynamic network settings and to obtain desirable global properties like resilience, scalability and adaptability without explicitly programming them into the individual agents. This represents a radical shift from traditional algorithmic techniques to that of obtaining desired system properties as a result of emergent behavior that often involves evolution, adaptation, and learning.

The dynamic network architectures that will be studied during the project include *peer-to-peer* (P2P) and *grid* computing systems [14, 3], as well as *ad-hoc networks* (AHN) [6]. P2P systems are distributed systems based on the concept of resource sharing by direct exchange between *peer* nodes, in the sense that all nodes in the system have equal role and responsibility [14]. Exchanged resources may include content, as in popular P2P document sharing applications, or CPU cycles and storage capacity, as in computational and storage grid systems. P2P systems exclude any form of centralized structure, requiring control to be completely decentralized. In AHN, heterogeneous populations of mobile, wireless devices cooperate on specific tasks, exchanging information or simply interacting informally to relay information between themselves and the fixed network [17]. Communication in AHN is based on multi-hop routing among mobile nodes. Multi-hop routing offers numerous benefits: it extends the range of a base station; it allows power saving; and it allows wireless communication, without the use of base stations between users located within a limited distance of one another.

We have chosen these domains for their practical importance in future distributed computing technologies as well as their potential for benefiting from our results. P2P/Grid systems can be seen as dynamic networking at the application level, while AHN results from dynamic networking at the system level. In both cases, the topology of the system typically changes rapidly due to nodes voluntarily joining or leaving the network, due to involuntary events such as crashes and network partitions, or due to frequently changing interconnection patterns.

The use of CAS techniques derived from nature in the context of information systems is not new. Numerous studies have abstracted principles from biological systems and applied them to network-related problems, such as routing. However, much of the current work in this area can be characterized as *harvesting* — combing through nature, looking for a biological system or process that appears to have some interesting properties, and applying it to a technological problem by modifying and adapting it through an enlightened trial-and-error process. The result is a CAS that has been empirically obtained and that appears to solve a technological problem, but without any scientific explanation of why.

BISON proposes to take exploitation of CAS for solving technological problems beyond the harvesting phase. We will study a small number of biology-inspired CAS, applied to the technological niche of dynamic networks, with the aim of elucidating principles or regularities in their behavior. In other words, BISON seeks to develop a rigorous understanding of why a given CAS does or does not perform well for a given technological problem. A systematic study of the rules governing good performance of CAS offers a bottom-up opportunity to build more general understanding of the rules for CAS behavior. The ultimate goal of the BISON project is then the ability to synthesize a

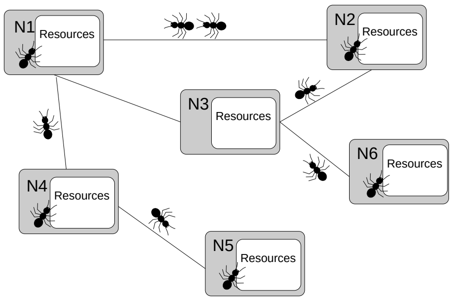

Fig. 1. An overlay network of nests in Anthill

CAS that will perform well in solving a given technological task based on the accumulated understanding of its regularities when applied to different tasks. In addition to this ambitious overall objective, BISON has more concrete objectives for obtaining robust, self-organizing and self-repairing solutions to important problems that arise in dynamic networks at both the system layer and the application layer.

BISON has officially started on January 1st, 2003, and will continue for three years. Given the interdisciplinary nature of the problem ahead, the BISON consortium brings together expertise from a wide range of areas including core disciplines (physics, mathematics, biology) and "user" disciplines (information systems, telecommunications industry). The BISON consortium consists of the Department of Computer Science at the University of Bologna (Italy), Telenor (the leading telecommunications operator in Norway), Department of Methods of Innovative Computing, University of Dresden (Germany), Dalle Molle Institute for Artificial Intelligence (IDSIA, Switzerland) and the Santa Fe Institute (USA).

3 The Anthill Framework

One of the goals of BISON is to develop a framework for the the design, analysis and evaluation of novel peer-to-peer applications based on ideas such as multi-agent systems borrowed from CAS [4]. This work will be based largely on the *Anthill* [1] framework that is being developed at Bologna as an open-source project. Anthill provides a "testbed" for studying and experimenting with CAS-based P2P systems in order to understand their properties and evaluate their performance. Although the current prototype of Anthill has been designed for P2P systems, it will serve as the basis for evaluation frameworks of other dynamic networks that will be studied in BISON.

An Anthill system is composed of a self-organizing overlay network of interconnected *nests*, as illustrated in Fig. 1. Each nest is a middleware layer capable of hosting resources and performing computations. The overlay network is characterized by the absence of any fixed structure, as nests come and go and discover each other on top of a communication substrate. Nests interact with local instances of one or more *applications* and provide them with a set of *services*. Applications are the interface between users and the P2P network, while services have a distributed nature and are based on the collaboration among nests. An example application might be a file-sharing system, while a service could be a distributed indexing service used by the file-sharing application to locate files.

An application performs *requests* and listens for *replies* through its local nest. Requests and replies constitute the interface between applications and services. When a nest receives a request from the local application, an appropriate service for handling the request is selected from the set of available services. This set is dynamic, as new services may be installed by the user. Services are implemented through *ants*, which are autonomous agents capable of traveling within the nest network. In response to a request, one or more ants are generated and assigned to a particular task. Ants may explore the nest network and interact with nests that they visit in order to accomplish their goal. Anthill does not specify which services a nest should provide, nor does it impose any particular format on requests. The provision of services and the interpretation of requests are delegated to ants.

Nests offer their resources to visiting ants through one or more *resource managers*. Example resources could be files in a file-sharing system or CPU cycles in a computational grid, while the respective resource managers could be a disk-based file system or a job scheduler. Resource managers typically enforce a set of policies for managing their (inherently limited) resources.

Ants do not communicate directly with each other; instead, they communicate indirectly by leaving information related to the service they are implementing at the appropriate resource manager found in the visited nests. For example, an ant implementing a distributed lookup service may leave routing information that aids subsequent ants in directing themselves toward regions of the network that are more likely to contain the searched key. This form of indirect communication, used also by real ants, is known as *stigmergy* [5].

A Java prototype of the Anthill runtime environment based on JXTA [9] has been developed [16]. JXTA, an open-source P2P project promoted by Sun Microsystems, aims to establish a programming platform for P2P systems by identifying a small set of basic facilities necessary to support P2P applications and providing them as building blocks for higher-level services. The benefits of basing our implementation on JXTA are several. For example, JXTA allows the use of different transport layers for communication and deals with issues related to firewalls and NAT.

In addition to the runtime environment, Anthill includes a simulation environment to help developers analyze and evaluate the behavior of their P2P systems. Simulating different P2P applications requires developing appropriate ant algorithms and a corresponding request generator characterizing user interactions with the application. Each simulation study is specified using XML by defining a collection of component classes and a set of parameters for component initialization. For example, component classes to be specified include the simulated nest network, the request generator to be used, and the ant algorithm to be simulated. Initialization parameters include the duration of the simulation, the network size, the failure probability, etc. This flexible configuration mechanism enables developers to build simulations by assembling pre-defined and customized component classes, thus simplifying the process of evaluating ant algorithms.

Unlike other toolkits for multi-agent simulation [12], Anthill uses a single ant implementation in both the simulation and actual run-time environments, thus avoiding the cost of re-implementing ant algorithms before deploying them. This important feature has been achieved by a careful design of the Anthill API and by providing two distinct implementations for it to be used in simulation and deployment.

Having developed a first Anthill prototype, we are now in the process of testing the viability of our ideas regarding biology-inspired approached to P2P by developing common P2P applications over it [1, 13]. As an example, we have developed Messor [13] which is a load-balancing application for grid computing that supports concurrent execution of highly-parallel, time-intensive computations where the workload may be decomposed into a large number of independent jobs. Messor exploits the computational power offered by a network of Anthill nests by assigning a set of jobs constituting a computation to a dispersed set of nests. To balance the resulting load among the nests, Messor uses a biology-inspired algorithm. Several species of ants are known to group objects (e.g., dead corpses) in their environment into piles so as to clean up their nests. It is possible to obtain this emergent behavior (randomly dispersed objects being transformed into neat piles) in a completely decentralized manner through virtual ants that act autonomously according to the following three rules [15]: (i) wander around randomly, until an object is encountered; (ii) if carrying an object, drops it and continue to wander randomly; (iii) if not carrying an object, pick it up and continue to wander randomly. Despite their simplicity, a colony of these "unintelligent" ants is able to group objects into large clusters, independent of their initial distribution in the environment.

The Messor algorithm is inspired by "inverting" the above virtual ant behavior such that an artificial ant drops an object it may be carrying only after having wandered about randomly "for a while" without having encountered other objects. Colonies of such ants try to disperse objects uniformly over their environment rather than clustering them into piles. As such, they could form the basis for a distributed load balancing algorithm.

Figure 2 illustrates the load balancing achieved by Messor over time. The results were obtained in a network of 20 Messor ants and 100 nests that are initially idle and form a ring-structured overlay network. At time zero, 10,000 jobs are inserted at a single nest. The different histograms depict the load observed at the 100 nests (linearly ordered along the x-axis) after 0, 5, 10, 15, 20, and 50 iterations of the algorithm. At each iteration, the 20 ants perform

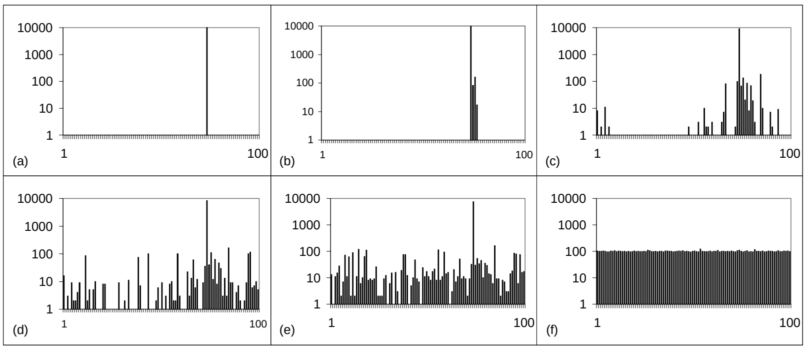

Fig. 2. Load distribution after 0, 5, 10, 15, 20, 50 iterations in 20-ant Messor.

a single step of the Messor algorithm: they randomly move from one node to another, possibly moving jobs with them from overloaded nests to underloaded nests. As the figure illustrates, after 15-20 iterations most of the nests in the network have some work to do, and after 50 iterations, the load is perfectly balanced. The first few iterations serve for the ants to explore their neighborhoods in the network and create new connections beyond the initial ring structure so that future steps of the algorithm can transfer load to more remote parts of the network.

4 Conclusions

BISON and Anthill are relatively new projects and results obtained so far are limited to a few "proof-of-concept" demonstration applications. Nevertheless, preliminary simulation results confirm our intuition that biology-inspired techniques for solving challenging problems in dynamic networks is indeed a promising approach. Our preliminary results exhibit performances that are comparable to solutions obtained through traditional techniques. Yet, the methodology followed to obtain our algorithms is completely different from previous approaches, as we mimic the behavior of natural systems, thus inheriting their resilience and self-organizational properties for free.

Additional information on Anthill and BISON can be found at their respective web sites `http://www.cs.unibo.it/projects/anthill/`, `http://www.cs.unibo.it/bison/`.

References

1. Ö. Babaoğlu, H. Meling, and A. Montresor. Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems. In *Proc. of the 22th Int. Conf. on Distributed Computing Systems*, Vienna, Austria, July 2002.

2. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.
3. I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a Future Computing Infrastructure,*. Morgan Kaufmann, 1999.
4. D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1999.
5. P. Grasse. La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. *Insectes Sociaux*, 6:41–81, 1959.
6. Z. Haas and M. Pearlman. The zone routing protocol (ZRP) for ad-hoc networks (InternetDraft). Mobile Ad-hoc Network (MANET) Working Group, Aug. 1998.
7. J. Holland. *Emergence: from Chaos to Order*. Oxford University Press, 1998.
8. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. of the Natl. Acad. Sci. USA*, 79:2554–2558, 1982.
9. Project JXTA. http://www.jxta.org.
10. S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
11. E. Klarreich. Inspired by Immunity. *Nature*, 415:468–470, Jan. 2002.
12. N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The Swarm Simulation System, A Toolkit for Building Multi-Agent Simulations. Technical report, Swarm Development Group, June 1996. http://www.swarm.org.
13. A. Montresor, H. Meling, and O. Babaoğlu. Messor: Load-Balancing through a Swarm of Autonomous Agents. In *Proc. of the 1st Workshop on Agent and Peer-to-Peer Systems*, Bologna, Italy, July 2002.
14. A. Oram, editor. *Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology*. O'Reilly, Mar. 2001.
15. M. Resnick. *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. MIT Press, 1994.
16. F. Russo. Design and Implementation of a Framework for Agent-based Peer-to-Peer Systems. Master's thesis, University of Bologna, July 2002.
17. C. Toh. *Ad Hoc Mobile Wireless Networks*. Prentice Hall, 2002.