New directions in the analysis of massive multi-agent systems

Charles L. Ortiz, Jr. and **Victoria Stavridou**
SRI International
Menlo Park, CA 94025
ortiz@ai.sri.com
stavridou@sdl.sri.com

Abstract

Distributed algorithms (that is, algorithms whose pieces run concurrently and independently on many, fallible, interconnected processors, each with a limited amount of information) for problems such as consensus, communication, resource allocation, and synchronization are well understood. This understanding is based on a foundation of impossibility results that define what algorithms can be used in what settings (defined by the timing model and the interprocess communication mechanism) and a set of precisely defined complexity measures that can be used to analyze how well the algorithms perform in these settings (Latombe 1991). Multi agent systems (distributed algorithms where the pieces are autonomous agents) are, by contrast, less explored. The dynamic properties of large-scale agent systems and the impact of collaboration and negotiation policies on global system behavior are not well understood. We have neither fundamental impossibility results nor well-defined complexity measures. For example, no systematic method exists for predicting whether particular negotiation approaches will converge, for estimating how long it will take and what the communication complexity will be, or for determining whether a multi agent system can deadlock or oscillate. The lack of these foundational results currently precludes the predictable deployment of agent solutions in applications with stringent performance constraints.

This paper outlines new directions in the development of foundational theories of multi agent systems algorithms. We explore the generalization of distributed systems theory results and complexity measures as well as two complementary approaches to analyzing the dynamic behavior of large-scale agent systems: qualitative analysis via proof and abstraction and quantitative estimation via statistical mechanics ideas such as phase transitions in combinatorial search problems.

Keywords: Large agent systems, resource management, statistical mechanics, formal properties.

Introduction

Many resource allocation problems can be usefully cast as problems involving the allocation of agents (resources) to tasks; in this paper we consider massive agent systems on the order of thousands of nodes, and discuss several new methods for studying the behavior of such systems. We restrict ourselves to *cooperative* agent systems in which each agent is working toward some common goal.

Given the localized behavior of a set of agents, A, it is important to ascertain whether certain behavior might emerge at the global level. Emergent behavior might be intended or unintended by the system designer and is analogous to program side effects. Examples of unintended harmful behavior include deadlock (agents cannot agree on a resource allocation), livelock, thrashing (one group of agents undoes the progress made by another group), divergence, lack of progress toward a goal (a group of agents is getting closer to a good resource allocation), etc.

There are a wide variety of mechanisms by which agents can interact in a multiagent system. A particular mechanism may be chosen depending on the goal that each agent or group of agents tries to achieve, and on environmental constraints. These mechanisms can be loosely grouped under the label of *negotiation* methods and cover a class of interaction protocols such as contract nets, auctions, and mediation-centered approaches.

Obstacles to the use of agent systems

In an agent system, control is local. Each agent is autonomous and behaves according to its own set of rules, and the interactions between agents are governed by these local rules. For such highly decentralized systems to be used in time-bounded resource allocation problems, it is important to understand the dynamics of the global system. Questions of convergence, stability, and communication and time complexity, must be answered. Whether or not a particular negotiation protocol converges is somewhat related to the question of whether or not two agents engaged in a negotiation using a particular protocol can expect to arrive at a "good enough" consensus after a certain number of rounds (S.Kirkpatrick & Selman 1994).

These problems are partially solved in special cases. For example, some negotiation approaches can be shown to converge to states that are optimal in some sense. However, these results typically apply only to small agent systems and often do not address important issues such as the time it can take for the system to converge or the impact of communication delays or failures. The problem of understanding or predicting with any degree of certainty the expected behavior of large-scale agent systems has not been solved in general, and this prevents such systems from being used in many applications where global failure can be catastrophic, such as safety-critical embedded systems.

In the absence of systematic techniques for predicting system dynamics, simulation and testing are the main validation approaches that can be used. Such approaches have well known limitations. In many distributed applications, network properties such as transmission delays can vary widely over time, and nodes and communication links can be unreliable. It is very difficult to simulate such environments faithfully. Furthermore, uncertainties about the environment often make testing results non-repeatable: for the same set of initial conditions, the same systems may behave in two completely different ways in test and in the real world. Even for well-controlled environments with predictable or constant characteristics, testing or simulation can only cover a very small sample of a typically infinite space of possible behaviors. Many conditions that the set of tests did no cover will be encountered in the field.

To enable the use of agent systems, we need better validation approaches than simulation or testing. We need systematic methods for investigating at design time the critical dynamic properties of an agent system. As far as possible, such methods should predict, under relevant assumptions about the environment, whether expected large-scale properties will hold with some degree of confidence. To complement such analytic approaches, it is also important to be able to detect at runtime that expected properties do not hold or that problems might occur, so that corrective actions can be taken.

New directions

To provide a sound basis to the design and deployment of agent systems, we discuss methods for both predicting and monitoring the global dynamic behavior of large agent systems. We discuss two complementary approaches: formal analysis via abstraction and proof, and probabilistic analysis using methods inspired from statistical mechanics.

Abstraction is a powerful means of verifying complex distributed systems: to show that a complex system S satisfies a property P, one constructs a simpler system S and a property P both derived from S and P in such a way that S satisfies P whenever S satisfies P. The practical interest of such an approach is that S can usually be much simpler than S, often simple enough that checking whether S satisfies P can be fully automatic. There are also algorithms for computing abstractions in many practical cases. We believe that existing abstraction techniques can be extended to large-scale agent systems. In particular, for the purpose of building monitors for agent systems.

Statistical mechanics studies the relation between microscopic and macroscopic properties of physical and chemical systems. Methods inspired from statistical mechanics have also been used to explain dependencies between large-scale and small-scale properties of computing applications. For example, the phase-transition phenomena that have been observed empirically in constraint satisfaction problems can be predicted theoretically using statistical models. Such models explain how simple parameters, such as the degree of the graph in graph-coloring problems or the number of clauses of a SAT problem, are good indicators of the hardness of particular problem and of the likelihood that the instance has solutions. A promising direction of research is the application of similar statistical models to large-scale agent systems to determine system and environment parameters, which provide statistically valid indication of the likelihood that some large-scale properties will be satisfied. We believe such results can be used for obtaining probabilistic guarantees about the dynamics of large agent systems, and for building runtime statistical monitors, capable of providing early indication that some property is likely to be violated.

The two approaches are complementary. Abstraction and proof can provide very strong guarantees about the dynamic behavior of a large agent system (e.g. absence of deadlock, convergence, or invariance properties) under various assumptions about the environment (e.g., delays on message transmission or bounds on the number of faulty nodes). Formal proof and abstraction are useful for showing that some property is satisfied for every possible system behavior. Statistical methods are useful when one wants to know the likelihood of a property holding rather than an absolute guarantee that it will be satisfied in all cases. Because proofs require all possible system evolutions to be considered, worst-case situations cannot be ignored however unlikely they might be. This can result in pessimistic results, far from what can be observed on average. For example, a proof that a particular system is guaranteed to converge to some equilibrium within a delay D does not give information about the average time for convergence, which may be much smaller than D. Statistical approaches on the other hand might give a good indication of the average delay to convergence but may not guarantee that the system will always converge.

Modeling and simulation of agent systems

One reasonable evaluation method for the approaches that we have outlined involve extensive simulation and experiment involving, for example, a set of benchmark problems from which a large set of problem instances can be automatically generated. This large set of problem instances could support empirical investigations on the average-case dynamic behavior of agent systems. The same set of problem instances could also serve to experimentally evaluate the theoretical results and the monitoring approaches.

For simulation and analysis purposes, one can model agents and the environment using appropriate types of automata. Automata representation is a classic approach to modeling distributed systems and algorithms, for which well-established verification and abstraction techniques have been developed. Automata have also been used to represent agents (Rushby 1998)

At SRI we have developed two systems, Maude (mau 1999) and ANCORS (Ricciulli 1998), for simulating agent systems and performing experiments. Maude is both a formal framework, built on declarative rewriting logic semantics, and an extremely efficient execution environment, achieving over a million rewrites per second on a cheap PC. Maude could be use for rapidly developing simulations of agent systems using rewrite rules to model agent evolution. ANCORS (Adaptive Network Controller and Reporting Systems) automates the assessment and control of changes to distributed environment assets. ANCORS runs on the target network itself. Through a hybrid approach combining simulation and emulation of the network, ANCORS automates the process of assessing and effecting a desired change to an active network.

Complexity measures

An obstacle to understanding the global properties of complex agent system is the lack of common measures for studying and comparing the performances of different approaches to negotiation. One natural starting point is the set of complexity measures that have been proposed and that are used to evaluate distributed algorithms. Examples of such measures include the number of rounds for an algorithm to terminate or the size and number of message exchanged. Various network properties or assumptions that are necessary for particular approaches to work should also be considered. Relevant network assumptions include the properties of the message delivery services (e.g., lossy vs. reliable channels, unbounded or bounded message transmission delays, or conditions on network topology and connectivity) as well as properties of nodes (e.g., presence or absence of clocks, reliability assumptions, or behavior in case of failure). These complexity measures and assumptions about the networking environment can provide a basis for evaluating and comparing negotiation protocols.

Analysis approaches

Abstraction and proof

Multi-agent systems are examples of large and complex distributed systems that cannot be analyzed thoroughly via testing. Instead, one can obtain very high assurance that such systems are correct via mathematically-based verification approaches: one builds a model of the system behavior that typically includes assumptions about the underlying communication network and shows that this model satisfies the algorithm requirements.

For systems that are sufficiently simple, specialized tools called model checkers can automatically determine whether critical properties are satisfied. Model checkers usually require the system under investigation to be modeled as a finite automaton and the verification is very much like an exhaustive exploration of all the possible runs of the automaton. In many cases however, the system to analyze is too complex or too large to be handled by model checkers. When the automatic techniques do not apply, the alternative approach is to construct a mathematical proof of correctness. Pen-and-paper proofs of distributed algorithms are notoriously error-prone and theorem-proving tools are often used to help the analysis and give assurance of correctness. However, even with such tools, verification tends to be tedious and lengthy, requiring heavy user interaction. Abstraction is a way to efficiently combine theorem proving and model checking. Abstraction techniques allow the extraction, from a concrete description of a system, of an abstract, therefore simplified, view of the system. If the transformation is well chosen, analysis at the abstract level is sufficient to establish that the concrete system satisfies the expected properties. Abstraction is especially effective if the abstract system is simple enough to be analyzed using model checking. To use such abstraction techniques, it is crucial to show that the properties of interest are preserved by the abstraction and this usually requires theorem proving. An abstract model is constructed manually and theorem proving is used to check that it is a proper abstraction of the concrete system and preserves the properties of interest. Recently, novel techniques based on abstract interpretation have been developed at SRI to compute automatically so-called boolean abstractions of transition systems (Glance & Huberman 1993). An abstract system computed in this way preserves, by construction, a large class of temporal properties of the original system. Theorem proving is used internally during the construction but in an automated way, without user intervention.

Many examples show that these techniques are very effective. A data-link protocol of industrial origin can be automatically verified in a few minutes using the abstraction and model checking tools included in PVS. In a previous study, verification of the same protocol using theorem proving alone took two months of effort.

At SRI we have been studying the application of abstraction approach to the dynamic behavior of multiagent systems. Our objective is to show that a system behavior satisfies global properties, given the local behavior of agents and the negotiation approach, using abstraction techniques. We believe that abstraction techniques will also be useful in addressing issues such as proving absence of deadlock or showing that negotiation protocols terminate within specified delays. For these purposes, we are studying the use of the SRI PVS system, a state-of-the-art theorem prover that integrates both model-checking tools and algorithms for computing abstractions.

Statistical approaches

Combinatorial search problems have easily computable characteristics that determine, on average, both the likelihood that they have solutions and their hardness, that is, the cost to solve them with a variety of heuristic search methods (S.Kirkpatrick & Selman 1994; Kephart, Hogg, & Huberman 1990). Important recent observations indicate abrupt transitions in behavior, analogous to physical phase transitions in percolation problems. In particular, these transitions are universal characteristics of classes of hard computational problems, independently of the choice of particular search algorithms. These results provide insight into the nature of NP-hard problems, but can they also lead to improved search methods? On the one hand, this seems likely in that all heuristics rely on approximate estimates of solubility and search cost associated with various choices to be made during the search. Thus the use of universal problem characteristics may usefully add global information to heuristics that evaluate choices based on limited, local views of the problem. However, the relation between these characteristics, and solubility and hardness only holds on average. The large observed variances indicate that any individual problem instance can deviate significantly from average behavior. This variance could easily render useless any reliance on average behavior.

This issue has been successfully resolved for some random search problems using genetic algorithms and backtracking. These applications require extending the theory of phase transition in search to apply to the situations encountered during search, and making specific choices on how to use the global information provided by the theory. The key insight allowing the theory to apply is that, at any point during a search, the remaining search is a subproblem of the original one. The theory can then be applied to this subproblem to suggest improvements in subsequent choices. These examples show how the underlying nature of problems can be used to devise better search methods. Even though the particular subproblems encountered during search depend on the previous choices made, treating these problems as if they were independently and randomly generated is accurate enough to give observable improvements in performance.

While these applications are encouraging, whether these techniques are useful for large real-world combinatorial searches remains to be seen. In particular, such search problems are likely to have characteristics (such as clustering of constraints) that differ significantly from typical random problems. Such differences can alter the location of phase transitions, so an algorithm based on random problems could incorrectly evaluate the likely search difficulty.

We believe that this problem can be addressed in the context of combinatorial searches involved in resource allocation in massive multiagent systems. In particular, it would be useful to examine how these transitions arise in combinatorial auctions and other protocols used to perform resource allocations, based on a variety of constraints on the agents.

Monitors derived from abstractions

Abstraction is traditionally used off-line to verify that a distributed system satisfies its requirements. These could also be used on-line to monitor whether an agent system satisfies or violates some global property. Abstractions can be useful for this purpose since they can be customized to capture properties of interest. An abstraction gives a simplified view of a global system where only the events affecting these properties are observed. An abstraction is a simplified model of the global system that is specific to a set of properties and whose evolution is determined by a set of abstract events relevant to these properties. By observing these abstract events at runtime and determining their effect on the abstract model, one obtains a runtime monitor in a natural way. The abstract model follows the global evolution of the agent system and its internal state reflects the current state of the global system, allowing it to detect and report the violation of expected properties.

This form of monitoring can rely on more general forms of abstraction than off-line verification. For verification purposes one strives to obtain a finite abstract model that can be analyzed via model checking. In a monitoring application, the abstract model does not need to be finite since no global space exploration is required. An interesting area of investigation is the development of techniques for building monitors for agent system from abstract models and generalizing existing notions of abstractions that work well for that purpose.

Statistical monitors

The traditional approach for observing the behavior of distributed system is event-based. The local activity of individual components (i.e. agents) produces sequences of primitive events that can be observed locally. These local event-sequences are then collected and analyzed in a hierarchical fashion: event reports from different nodes are assembled by monitors that perform filtering, combine low-level events to form more abstract ones, and distribute these more abstract events to higher-level monitors. Ultimately, event reports are sent to human operators or network management tools. At the center of this approach are techniques for building meaningful statistics about system activity.

We believe such a hierarchical monitoring approach can be applied to large-scale event systems. In particular, we believe that the results of the research on statistical analysis of agent systems discussed earlier could be used to determine which system parameters must be measured and how the local measures must be combined to provide valid indication that large-scale dynamic properties are likely to be violated.

A further promising area of research involves the investigation of mappings between such monitoring problems and constraint satisfaction problems (CSP) so as to use known results on phase-transition phenomena in this context. It may be possible to characterize certain global states (e.g. a global deadlock) as solutions of a set of local constraints on the internal state of agents and on environment variables (e.g. each agent is in a waiting state and no environment stimulus is present). Each solution of the set of constraints could be represented as a global system state that satisfies a desirable or undesirable property. Under the assumption that the larger the set of solutions, the more likely the system is to reach such a global state, the likelihood of the property being satisfied is related to the probability of the set of constraints having many solutions. Results from the study of phase transition in CSP show that this probability can be estimated, on average, by simple system parameters. Monitoring can then be based on continuously measuring these parameters. The validity of this approach could be evaluated experimentally, or using statistical models, for different classes of problems.

Just like results on the statistical analysis of combinatorial problems have led to new heuristics and approaches to solving constraint satisfaction problems, we can expect the proposed research on abstraction and statistical analysis of agent systems to provide new insights in the dynamic behavior of such systems. In may be possible to exploit these by defining system requirements concerning both negotiation protocols and system structure that can ensure either that some high-level property is satisfied, or that the system evolution will be easily monitored at runtime.

Summary

We have described several new promising directions for research in massive agent-based systems. Such agent systems could find application in areas such as communications systems, ad-hoc networks, battlefield networks, and teams of autonomous vehicles. The approaches we have considered combine ideas from a number of disciplines: (1) proof-based approaches from formal methods in computer science, (2) statistical mechanics methods from physics as applied to combinatorial search problems, and (3) agent-based systems in AI.

References

Glance, N. S., and Huberman, B. A. 1993. Organizational fluidity and sustainable cooperation. In *Proc. of MAAMAW-93*.

Kephart, J. O.; Hogg, T.; and Huberman, B. A. 1990. Collective behavior of predictive agents. *Physica* D42:48–65.

Latombe, J. 1991. *Robot Motion Planning*. Academic Publishers.

1999. Maude: Specification and programming in rewriting logic. Technical report, SRI International.

Ricciulli, L. 1998. High-fidelity distributed simulation of local area networks. In *31st Annual Simulation Symposium*.

Rushby, J. 1998. Ubiquitous abstraction: a new approach for mechanized formal verification. In *Second IEEE International Conference on Formal Engineering Methods*, 176–178.

S.Kirkpatrick, and Selman, B. 1994. Critical behavior in teh satisfiability of random boolean expressions. *Science* 264:1297–1301.