Cooperative Computing in Dynamic Environments
MIT9904-12

Progress Report: January 1, 2000—June 30, 2000

Nancy Lynch and Idit Keidar

A. Project Overview

We are working on developing models and analysis methods for distributed systems, with a focus on cooperative group activities in networks. Such group activities range from human social activities in cyber communities to powerful distributed applications involving data sharing and cooperative work. These activities are often supported by agent communication services, which provide distributed intelligence, or by group communication services, which manage group membership and guarantee coherent communication. The environments in which such activities take place are highly dynamic: participants come and go (and change location), network topology changes, and components fail and recover. Coping with such difficult environments leads to complex implementations, which are difficult to build, understand, and analyze.

This project addresses these problems using formal modeling and verification techniques, in particular, a combination of Input/Output automaton methods used at MIT and process algebraic and knowledge-based methods used at NTT. This involves extensions to the existing techniques, for example, extending I/O automata to allow dynamic process creation and destruction. As the basic framework is developed, it is being applied to a collection of typical examples from cooperative computing applications, including computer-supported cooperative work, e-commerce, and distributed databases. Other issues being studied include analysis of performance and fault-tolerance properties, and connecting the formal models with actual runnable code.

B. Progress through June, 2000:

B1. Agents

We have developed a new dynamic I/O automaton (DIOA) model, which extends the basic I/O automaton model to allow dynamic automaton creation and destruction and changes to an existing automaton's signature. In particular, the model allows changes in the actions that may be used for communication with other automata. We have also outlined how this model can be used to support description of agent mobility.

The current version of the model includes certain complications, such as the possibility of creating indistinguishable automaton "clones". Work remains to remove some of these complications.

We have completed our comparative case study of three formal methods for modeling and analyzing agent programs, namely, knowledge-based programming (Erdos), a process-algebraic method (NePi2) and dynamic I/O automata. The case study involves a simple e-commerce example. We have presented this case study at a NASA workshop on formal techniques for agent systems. We have produced a preliminary correctness proof for the I/O automaton version of the case study.

Dr. Kawabe of NTT has visited MIT. He wrote an interpreter for most of NePi2 (all the significant constructs except for nondeterministic choice) using IOA. Also, at the end of this reporting period, Profs. Lynch and Attie visited NTT. They worked on DIOA and its extensions, applications of IOA and DIOA to analyze NePi2 and Erdos programs, and modeling of agent primitives using DIOA.

B2. Group Communication

During this reporting period we have continued our efforts in the area of group communication systems. We have continued our efforts towards building group communication services for WANs. We are working to prepare journal versions of the papers [KK00,KSMD00].

In [KSMD00], we describe a novel scalable group membership algorithm for WANs. Our membership service does not evolve from existing LAN-oriented membership services; it was designed explicitly for WANs. Our membership service is scalable in the number of groups supported, in the number of members in each group, and in the topology each group spans. Our service also supplies the hooks needed to provide clients with full virtual synchrony semantics.

Our service attains, on average, a low message overhead by agreeing on membership within a single message round. It avoids flooding the network and uses a scalable failure detection service designed for WANs. Furthermore, our service avoids notifying the application of obsolete membership views when the network is unstable, yet it converges when the network has stabilized. In contrast to most group membership services, we separate membership maintenance from reliable communication in multicast groups: membership is not maintained by every process, but only by dedicated servers.

In the past months, we have implemented the service in C, using the WAN-oriented failure detection service of [ABDL97]. We are currently running the service over the Internet at several locations around the world -- in several locations in the US, a location in Israel, and one in Taiwan. We are studying and tuning the service performance.

In [KK00], we present a novel design for a novel Virtually Synchronous group communication service targeted for WANs. We are currently completing the journal version of this paper. In the full paper, we make the following contributions:

- We design a new algorithm for implementing Virtual Synchrony. Our algorithm is more efficient than existing algorithms. It neither processes nor delivers obsolete membership views. Moreover, the synchronization protocol run by our algorithm involves just a single message exchange round among members of the new view.

- Our design demonstrates how to more effectively decouple the algorithm for achieving Virtual Synchrony from the algorithm for maintaining membership. Such efficient decoupling is important for providing scalable group

communication services in WANs. Our design allows the membership algorithm to freely change memberships of forming views at any time. The interaction between the membership and Virtual Synchrony algorithms is only one-way, from the former to the latter, and it has low overhead. The decoupling is such that the synchronization protocol can execute in parallel with the view formation protocol.

- Our design is carried out much more rigorously and formally than previous designs of Virtually Synchronous group communication services. The presented specifications of our service and its environment, description of the algorithm, and proof of correctness are all precise and formal.

In order to manage the complexity of the design, we have developed a novel, inheritance-based, methodology [KKLS00]. This methodology allows for incremental construction of formal specifications, models, and very importantly proofs. In addition to making the design tractable, the use of this methodology makes it evident which part of the algorithm implements which property and why.

In [BKAL00] we present an algorithm for totally ordered multicast which preserves Quality of Service (QoS) guarantees. We assume a QoS reservation model in which the network allows for reservation of variable bandwidth, specified by the average transmission rate and the maximum burst. As long as the application sends at the reserved rate, the network guarantees to deliver messages with bounded delays. For this model, we present a totally ordered multicast algorithm that preserves the bandwidth and latency reserved by the application within certain additive constants. The algorithm allows for dynamic joining and leaving of processes while still preserving the QoS guarantees.

In [Ingols00] we present a framework for the implementation of primary component algorithms. This framework is used to implement several algorithms based on the dynamic voting principle. We then study the algorithms, using simulations. The study shows that such algorithms are significantly affected by factors which have been overlooked in previous studies of dynamic voting availability. Specifically, we show that an algorithm's performance is highly affected by interruptions; availability degrades as more connectivity changes occur, and as these changes become more frequent.

De Prisco's new group communication service, a "dynamic configuration" service, is designed to tolerate both transient faults (using quorums or other structure associated with a view of the group) and longer-lasting system configuration changes (using a reconfiguration protocol). His thesis was completed in December 1999; during the current reporting period, we prepared a conference submission.

C. Plans for the next 6 months:

C1. Agents

We will improve the DIOA model to remove some of the complications, in particular, we will remove clones and will restrict primitives such as creation so that they may be analyzed using standard IOA compositional methods. We will write a conference paper on this model. We will consider adding extra features such as timing to DIOA. We will attempt to view the addition of dynamic features such as create, destroy, etc. as applying transformations to a basic static model (which might include timing).

In terms of the improved DIOA model, we will formulate fundamental primitives of agent programming languages, for example, the meet and connect primitives of Telescript.

We will help Dr. Mano and Dr. Kawabe of NTT to model and analyze a distributed implementation of NePi^2 using IOA. The key aspect to model/analyze appears to be the strategy used for scheduling pairwise communications along channels. This project will build on Dr. Kawabe's preliminary interpreter for NePi2 using IOA; it will also help in extending that interpreter to include NePi2's nondeterministic choice construct.

We will also consider additional applications of our agent framework, perhaps derived from the area of e-commerce. We will assist Dr. Araragi of NTT in modeling and analyzing a particular agent application, involving the execution of downloaded code.

C2. Group communication

In the next six months, we intend to continue the above research, focusing on performance and implementation of the Virtual Synchrony algorithm above [KK00]. Specifically, we are working on implementing the algorithm in C++, and we intend to formally analyze its performance. In addition, we intend to introduce optimizations to the algorithms to achieve better performance. We also intend to finalize journal versions of the papers [KK00,KSMD00].

Our work on group communication services is aimed at providing middleware support for WAN applications that require a certain degree of consistency, mainly collaborative computing applications such as drawing on a shared white-board or a shared text editor. In the coming months, we intend to study alternative approaches to building middleware support for similar applications.

One such approach can be providing totally ordered multicast services that preserve Quality of Service (QoS). We are now starting to study the QoS guarantees of totally ordered multicast algorithms. In [BKAL00], we have presented QoS-preserving totally ordered multicast. In the next months, we are striving to construct an algorithm for Atomic Broadcast which would also preserve Quality of Service guarantees.

We also intend to study scalable reliable multicast algorithms built over an underlying unreliable multicast such as IP-Multicast.

We will also define specifications for new services (e.g., resource allocation services, consensus services) suitable for environments in which participants can come and go. We are interested in determining reasonable correctness requirements for such services, as well as the costs of implementing such services.

References:

[KK00] Idit Keidar and Roger Khazan. A Client-Server Approach to Virtually Synchronous Group Multicast: Specifications and Algorithms. 20th International Conference on Distributed Computing Systems (ICDCS), April 2000. To appear. MIT Lab. for Computer Science Tech. Report MIT-LCS-TR-794.

[KSMD00] I. Keidar and J. Sussman and K. Marzullo and D. Dolev. A Client-Server Oriented Algorithm for Virtually Synchronous Group Membership in WANs. 20th International Conference on Distributed Computing Systems (ICDCS), April 2000. To appear. Full version: MIT Technical Memorandum MIT-LCS-TM-593.

[ABDL97] T. Anker and D. Breitgand and D. Dolev and Z. Levy. *Congress* CONnection-oriented Group-address RESolution Service. Proceedings of SPIE on Broadband Networking Technologies, Dallas, Texas, 1997.

[KKLS00] Idit Keidar and Roger Khazan and Nancy Lynch and Alex Shvartsman. An Inheritance-Based Technique for Building Simulation Proofs Incrementally. 22nd International Conference on Software Engineering (ICSE), Limerick, Ireland, pages 478-487, June 2000.

[BKAL00] Ziv Bar-Joseph and Idit Keidar and Tal Anker and Nancy Lynch. QoS Preserving Totally Ordered Multicast. Technical report MIT-LCS-TR-796, MIT Laboratory for Computer Science, January 2000. Url: http://theory.lcs.mit.edu/$\sim$idish/Abstracts/qos.html}

[Ingols00] Kyle W. Ingols. Availability Study of Dynamic Voting Algorithms. Master of Engineering, Department of Electrical Engineering and Computer Science, MIT, May 2000. Url http://theory.lcs.mit.edu/~idish/Abstracts/ingols-thesis.html.