Mailbox-Based Scheme for Mobile Agent Communications

The authors present a flexible and adaptive scheme that associates each mobile agent with a mailbox but lets them decouple. Their 3D model provides a basis for evaluating existing communication protocols and allows for the design of new ones to meet various application requirements.

Jiannong Cao
The Hong Kong Polytechnic University

Xinyu Feng and Jian Lu
Nanjing University

Sajal K. Das
University of Texas at Arlington

Mobile agent technology has great potential for use in networking and distributed systems. Its applications range from telecommunications, e-commerce, and information searching to process coordination, mobile computing, and network management.[1] Mobile agents are autonomous objects or object clusters that move between locations in a mobile agent system—a distributed abstraction layer that provides the concepts and mechanisms for mobility and communication.[2,3]

Communication protocols are among the most important mechanisms in mobile agent systems.[4] In various situations, mobile agents at different hosts must cooperate with one another by sharing information and making decisions collectively.[5] To ensure effective interagent communication, these protocols must track target agent locations and deliver messages reliably.

In recent years, as the "Mobile Agent Tracking and Message Delivery Protocols" sidebar describes, researchers have proposed a wide range of schemes for agent tracking and reliable message delivery. However, each scheme has it own assumptions, design goals, and methodology. Protocol requirements with respect to different classes of applications are not well understood. As a result, no uniform or structured methods exist for characterizing current protocols, making it difficult to evaluate their relative effectiveness and performance.

In response to this problem, we propose a mailbox-based scheme for designing mobile agent communication protocols. This scheme assigns each agent a mailbox to buffer messages but decouples the agent and mailbox to allow them to reside at different hosts and migrate separately.

On the basis of the scheme's design space, we have developed a three-dimensional model that captures the main features of any communication protocol. This model provides a basis for evaluating various existing mobile agent communication protocols and for helping users design a flexible, adaptive protocol they can customize to meet specific application requirements.

PROTOCOL REQUIREMENTS

Communication protocols for mobile agents require location transparency, reliability, efficiency, asynchrony, and adaptability.

Location transparency

Because mobile agents can move autonomously from host to host, they cannot reliably "know" the locations of their communication peer. Therefore, a practical communication protocol must keep track of agent locations, allowing each agent to send messages to its peers without knowing where they physically reside.

0018-9162/02/$17.00 © 2002 IEEE

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on March 16, 2009 at 03:20 from IEEE Xplore. Restrictions apply.

Mobile Agent Tracking and Message Delivery Protocols

Researchers have recently proposed many mobile agent tracking protocols in different contexts ranging from mobile and wireless communications to wide-area distributed systems.[1] These protocols use various approaches that typically rely on some combination of a home server, forwarding pointers, broadcasts, and a hierarchical location directory.

Home server

Several mechanisms use a location server to keep track of a mobile object's current location. For example, the mobile Internet protocol uses a *home server* to route IP packets. A mobile host registers its care-of-address with its home host every time it moves. The home server directs all IP packets to the home host, which forwards them to the mobile one.

Mobile agent systems also use the home-server protocol as proposed in the Object Management Group's Mobile Agent System Interoperability Facility.[2] The protocol is simple to implement. Locating a mobile object incurs little communication overhead, but updating the locations and delivering messages incurs large costs.

To avoid a triangular routing problem, some researchers have proposed cache-based strategies.[3] The Internet mobile host protocol forwards packets along pointers left by the mobile host if a cache miss occurs.[4] However, these low-level protocols do not handle message loss due to host mobility.

Forwarding pointers

In some tracking mechanisms,[5] each host on a mobile agent's migration path keeps a *forwarding pointer* to the next host on the path. Each sender knows the target agent's home. Messages are sent to the agent's home and forwarded to the target object along the forwarding pointers. A path compression technique reduces message hops. After routing a message to the target object, the system sends an `Update_Entry` message back along the chain and updates forwarding pointers in the chain's nodes.

Other approaches

Many tracking protocols use broadcast and hierarchical approaches to implement distributed location management.

In a *broadcast* approach, the message sender broadcasts both location queries and pending messages to all system hosts.

However, simple broadcasts cannot avoid message loss that object mobility causes. A snapshot broadcast strategy[6] could be used to guarantee reliable message delivery to highly mobile agents as well as for group communications, but the large overhead makes broadcasts impractical in large-scale networks.

In the *hierarchical* approach,[7,8] a hierarchy of servers forms a location directory. The location database at each level contains location information for objects at levels below it. The hierarchy is usually tree-structured, with a leaf containing entries for all mobile objects in the corresponding unit zone. An internal node maintains data about mobile objects registered in its subtree's set of zones. For each object, the information is either a pointer to an entry at a lower-level location server or the object's actual current position.

Adaptive location management

A proposed *update* strategy for personal communication services optimizes location management cost on a per-user basis.[9] Each mobile user has an update strategy based on a unique mobility model and call arrival pattern, consisting of a set of binary decision variables—to update or not—for all areas. Users who enter a new location area can choose whether to update this information.

Different mobility patterns have prompted other update strategies based on timers, movement, distance, and state.[10]

Message delivery

Resending-based protocols[11] guarantee reliable message delivery to mobile agents. Using sliding-window mechanisms similar to those in the Internet transmission-control protocol, the sender can detect the loss of a message and resend it. After several resendings, the sender contacts the location manager and delivers the message to the receiver's new address. For highly mobile agents, there is no upper limit to the number of message resendings. Implementing reliable message delivery, however, requires synchronizing the message-passing operation with agent migration.

References

1. E. Pitoura and G. Samaras, "Locating Objects in Mobile Computing," *IEEE Trans. Knowledge and Data Eng.,* vol. 13, no. 4, 2001, pp. 571-592.
2. D. Milojicic et al., "MASIF: The OMG Mobile Agent System Interoperability Facility," *Proc. 2nd Int'l Workshop Mobile Agents* (MA 98), Springer-Verlag, Berlin, 1998, pp. 50-67.
3. Y. Lin, "Determining the User Locations for Personal Communications Services Networks," *IEEE Trans. Vehicular Technology,* vol. 43, no. 3, 1994, pp. 466-473.
4. C. Perkins, A. Myles, and D.B. Johnson, "IMHP: A Mobile Host Protocol for the Internet," *Computer Networks and ISDN Systems,* vol. 27, no. 3, 1994, pp. 479-491.
5. J. Desbiens, M. Lavoie, and F. Renaud, "Communication and Tracking Infrastructure of a Mobile Agent System," *Proc. 31st Ann. Hawaii Int'l Conf. System Sciences* (HICSS 98), IEEE CS Press, Los Alamitos, Calif., 1998, pp. 54-63.
6. A.L. Murphy and G.P. Picco, "Reliable Communication for Highly Mobile Agents," *Proc. 1st Int'l Symp. Agent Systems and Applications and 3rd Int'l Symp. Mobile Agents* (ASA/MA 99), IEEE CS Press, Los Alamitos, Calif., 1999, pp. 141-150.
7. M. van Steen et al., "Locating Objects in Wide-Area Systems," *IEEE Comm. Magazine,* vol. 36, no. 1, 1998, pp. 104-109.
8. W.V. Belle, K. Verelst, and T. D'Hondt, "Location Transparent Routing in Mobile Agent Systems—Merging Name Lookups with Routing," *Proc. 7th IEEE Workshop Future Trends of Distributed Computing Systems* (FTDCS 99), IEEE CS Press, Los Alamitos, Calif., 1999, pp. 207-212.
9. S.K. Sen, A. Bhattacharya, and S.K. Das, "A Selective Location Update Strategy for PCS Users," *ACM/Baltzer J. Wireless Networks,* vol. 5, no. 5, 1999, pp. 311-326.
10. V.W-S. Wong and V.C.M. Leung, "Location Management for Next-Generation Personal Communication Networks," *IEEE Network,* vol. 14, no. 5, 2000, pp. 18-24.
11. M. Ranganathan, M. Bednarek, and D. Montgomery, "A Reliable Message Delivery Protocol for Mobile Agents," *Proc. Joint 2nd Int'l Symp. Agent Systems and Applications and 4th Int'l Symp. Mobile Agents* (ASA/MA 00), Springer-Verlag, Heidelberg, Germany, 2000, pp. 206-220.

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on March 16, 2009 at 03:20 from IEEE Xplore. Restrictions apply.

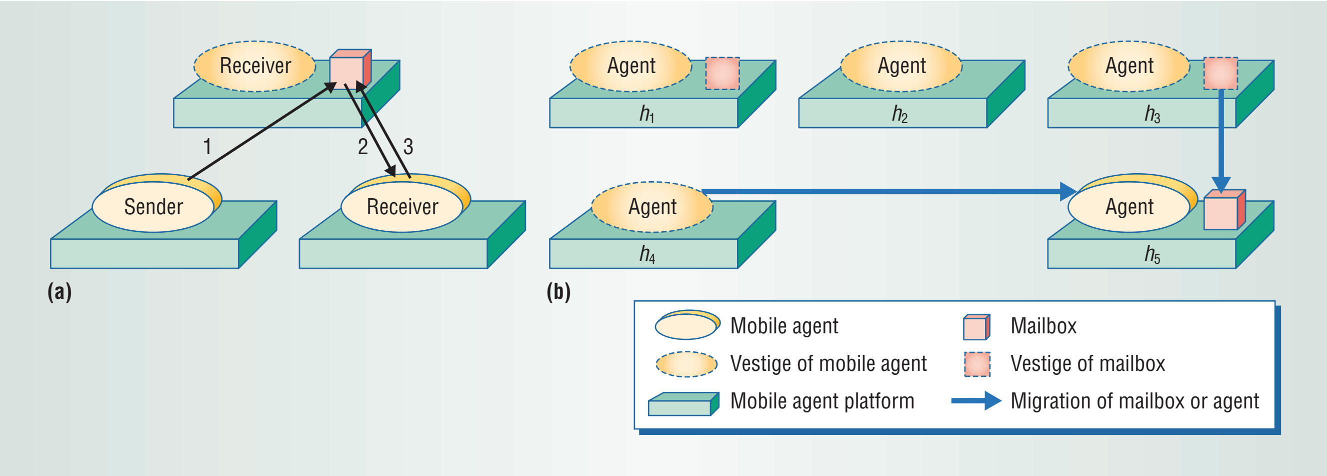

Figure 1. Mailbox-based scheme. (a) An agent sends a message to another agent's mailbox (1), from which the receiver later obtains it via either a (2) push or (3) pull operation. (b) A mobile agent's mailbox migrates at a lower frequency than the agent itself.

Reliability

The asynchronous message passing and agent migration in mobile agent systems can result in message loss. Even an ideal fault-free transport mechanism cannot guarantee the delivery of messages to their destination agents. A practical protocol, however, must ensure that all messages are routed to the target agent, no matter how frequently it migrates, in a bounded number of hops.

Efficiency

Protocol costs include the number of messages sent, their size, and the distance traveled. An efficient protocol minimizes these costs while supporting two operations: agent migrations to new sites and message delivery, which includes first locating the target agent. However, minimizing the costs of these operations leads to conflicting requirements.

Consider the tradeoff between a *full-information* strategy in which every site in the network maintains complete up-to-date information about every agent's location, and a *no-information* strategy that does not update agent migration information.[6] Delivery is inexpensive with the former approach, but updating location information at every site makes migration expensive. With the latter strategy, on the other hand, migration costs nothing, but delivery requires searching the entire network.

Any protocol must balance these costs either generally or within some specific communication and migration pattern.

Asynchrony

Although a communication protocol should coordinate message forwarding with agent migration to guarantee reliable message delivery, it should not overly constrain agent mobility by frequent and tight synchronization. Agents should be able to freely migrate to other hosts whenever necessary.

In addition, the agent's autonomous and asynchronous execution should not rely heavily on the agent's home to locate and deliver every message to them. Agents should be independent of the process that created them, and the home site should be able to disconnect from an agent as soon as it migrates.

To retain the advantages of asynchrony in mobile agent systems, any communication protocol must address asynchronous requirements in both agent migration and execution.

Adaptability

Different applications often have their own requirements. Some, for example, favor asynchrony, while reliability is more important in others. Various interagent communication and agent migration patterns also may have unique implications on migration and delivery strategies. Protocols can be designed for specific applications to achieve optimal performance, but an adaptive protocol that can suit as many kinds of applications as possible in a general-purpose mobile agent system is more desirable.

MAILBOX-BASED SCHEME

In our scheme, each mobile agent has a mailbox that buffers the messages sent to it. As Figure 1a shows, an agent can direct a message to another agent's mailbox, and the receiving agent uses a push or pull operation to obtain the message from the mailbox.

Although logically part of the agent, the mailbox can be detached from its owner—the agent can leave the mailbox at one host as it migrates to a new one. Interagent communication thus consists of two distinct steps:

- transmission of a message from the sender to the receiver's mailbox, and
- delivery of the message from the mailbox to its owner agent.

Mailboxes are mobile objects, but because they are not autonomous, they cannot determine their migration path. Any existing message delivery strategy will satisfy the first step, but because mailboxes migrate at a much lower frequency than their associated agents, any protocol design must include a

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on March 16, 2009 at 03:20 from IEEE Xplore. Restrictions apply.

parameter for when to move the mailboxes.

Figure 1b shows an agent migrating sequentially from host h_1 to h_5 but taking its mailbox only while moving to h_1, h_3, and h_5. By definition, the set of hosts on the mailbox's migration path is a subset of the hosts on the mailbox owner's migration path. The mobile agent's home is the first host on the migration paths of both the agent and its mailbox.

Because it is rare for two mobile agents roaming the Internet to use synchronous communication, we assume that mobile agent communication is largely asynchronous. The Internet's lengthy and unpredictable message delays, which can easily last several seconds, likewise prohibit frequent use of synchronous communication in a mobile agent application.

Our scheme is built on top of a reliable network communication layer, which guarantees that messages will not be lost during transmission and will be delivered between hosts.

DESIGN OPTIONS

To meet an application's specific requirements, our scheme offers choices in three aspects of protocol design: mailbox migration frequency, mailbox-to-agent message delivery, and synchronization of message forwarding with object migration.

Mailbox migration frequency

The number of mailbox migrations during a mobile agent's life cycle, along with the times when the migrations occur, can vary.

No migration (NM). A mobile agent can move alone, leaving its mailbox at home during the agent's life cycle. All messages are sent to the home, and the agent uses a mailbox-to-agent delivery mode to obtain messages. Tracking the mailbox carries no cost, but the agent's home must forward all messages. This triangular routing[7] increases the communication overhead for message delivery.

Full migration (FM). As part of the mobile agent's data, the mailbox continuously migrates with the agent. Although the cost of message delivery between mailbox and agent is zero, tracking the mailbox is difficult. Frequent agent-mailbox migration necessitates a tradeoff between message loss and the cost of guaranteeing message delivery.

Jump migration (JM). A mobile agent can determine dynamically whether to take its mailbox before each migration. In making the decision, the agent may consider the number of messages it will receive at its target host, the distance between the target host and the host where its mailbox currently resides, and other factors.

An agent that seldom receives messages at its target host doesn't need to take its mailbox to the new host. However, if an agent expects to receive messages frequently and its target host is far from the host where its mailbox currently resides, fetching messages from the remote mailbox will be expensive. In this case, the agent should migrate with its mailbox to the target host.

In JM mode, a protocol can work more flexibly when based on a decision suited to particular agent migration and interagent communication patterns, reducing the cost of both tracking and delivery operations.

The Internet's unpredictable message delays prohibit frequent use of synchronous communication in a mobile agent application.

Mailbox-to-agent message delivery

Messages destined to an agent are all sent to the agent's mailbox, and the agent later receives the messages by either a push or pull operation.

Push (PS). The mailbox keeps its owner agent's address and forwards every message to it. Although message queries incur no costs, the agent must notify the mailbox of its current location after every migration. If the agent migrates frequently but communicates with other agents at only a few hosts on the migration path, most message delivery location registration messages would be superfluous and introduce a significant migration overhead. PS mode is needed if real-time message delivery is required.

Pull (PL). The mobile agent retrieves messages from its mailbox's address whenever needed. The mailbox doesn't need to know the agent's current location, thus avoiding location registration, but the agent must query its mailbox for messages. Polling messages would increase message delivery overhead.

Migration-delivery synchronization

Our scheme lets users determine whether they need reliable message delivery. If users require high reliability, they can overcome message loss by

- synchronizing the *host*'s message forwarding and the *mailbox*'s migration (SHM),
- synchronizing the *mailbox*'s message forwarding and the *agent*'s migration (SMA), or
- both, known as full synchronization (FS).

NS denotes the extreme case of no synchronization performed.

Figure 2 shows how synchronization occurs between the message-forwarding object (mobile agent server or mailbox) and the moving object

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on March 16, 2009 at 03:20 from IEEE Xplore. Restrictions apply.

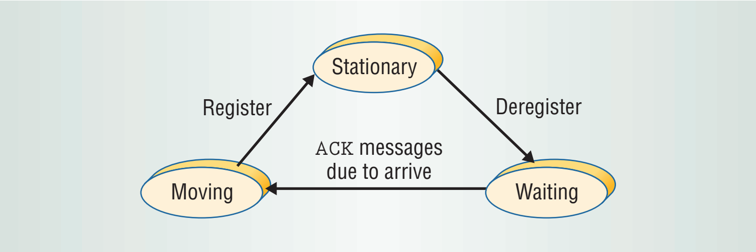

Figure 2. State switching of a mobile object. During synchronization, the communication protocol can forward messages to the mobile object in either a stationary or waiting state, but it must block messages when the object is moving.

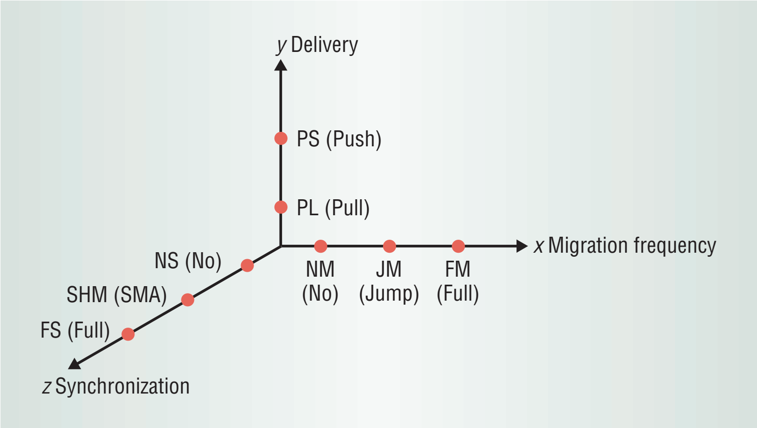

Figure 3. Three-dimensional design space. Each axis represents one design aspect and contains a range of properties that designers can combine with properties from other dimensions in various ways.

Table 1. Parameter combinations and corresponding protocols.

Protocol	Location registration	Reliability
NM-PS-NS	Yes (agent → mailbox)	No
NM-PS-SMA	Yes (agent → mailbox)	Yes
NM-PL-NS	No	Yes
FM-*-NS	No	No
JM-PL-NS	No	No
JM-PS-NS	Yes (agent → mailbox)	No
FM-*-SHM	Yes (mailbox → host)	Yes
JM-PL-SHM	Yes (mailbox → host)	Yes
JM-PS-FS	Yes (agent → mailbox, mailbox → host)	Yes

(mailbox or mobile agent). Before migrating, the moving object sends `Deregister` messages to all objects that might forward messages to it and waits for `ACK` messages from each forwarding object, which contains the number of messages to be forwarded. Once the messages arrive, the moving object continues on its path. After migration, the moving object informs all message-forwarding objects of its arrival by sending them `Register` messages.

3D DESIGN MODEL

These three design options generate a three-dimensional design space. As Figure 3 shows, each option represents one orthogonal dimension with a constraint spectrum. Because the three dimensions are independent of one another, designers can combine properties from different dimensions in various ways. The full range of properties for a given application can thus vary greatly.

Combining parameters from all three dimensions yields a taxonomy of mobile agent communication protocols. A string of the format XX-YY-ZZ expresses a protocol in which XX represents mailbox migration frequency (NM, JM, or FM), YY stands for mailbox-to-agent message delivery (PL or PS), and ZZ symbolizes synchronization (NS, SHM, SMA, or FS). A protocol's overall configuration has a special value for each of the three parameters. Most combinations have plausible applications.

Table 1 shows the different protocols derived from our mailbox-based scheme, with the description of their location registration modes and whether they can satisfy the required reliability. An asterisk denotes that multiple values are applicable.

Home-server protocols

All NM-mode protocols adopt a home-server approach. In this case, the agent's home acts as the message-forwarding server.

NM-PS-NS is identical to mobile IP.[7] The agent registers its current location with its mailbox residing at its home. Messages are sent to the mailbox, which pushes them to its owner agent. This protocol does not guarantee message delivery: If the agent migrates during message forwarding, the message will be lost.

Reliable message delivery requires synchronizing agent migration and message forwarding from the mailbox. This results in NM-PS-SMA, a synchronized version of mobile IP. In NM-PL-NS, the agent pulls messages from its mailbox, ensuring message delivery without using synchronization.

Home-server protocols are simple and work well for small-to-medium systems with relatively few agents. However, triangular routing increases communication overhead, especially when agents are widely distributed. In a system with numerous agents and frequent interagent communication, the home may become a performance bottleneck and a single point of failure. Further, mobile agent dependence on the home as a message-forwarding server constrains asynchronous execution.

Forwarding-pointer-based protocols

In FM-*-NS, each host on the mailbox migration path keeps a forwarding pointer to the successive host. The sender caches the target mailbox's previously obtained location; if the cache has no such

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on March 16, 2009 at 03:20 from IEEE Xplore. Restrictions apply.

address, the sender uses the target agent's home as the mailbox's cached address. Messages are sent to the cached address directly; if a cache miss occurs, the messages are forwarded along the pointers.

When the mailbox receives the message and discovers that the sender has outdated knowledge of its address, the mailbox notifies the sender of its current location, and the sender updates the cached address. Because the mailbox is bound with its owner agent in the FM mode, no remote interaction occurs between the mailbox and agent.

JM-PL-NS and JM-PS-NS are similar to FM-*-NS except that the mailbox migrates in JM mode, which is a kind of path compression technique. When the agent and its mailbox are at different hosts, the agent uses the pull and push operations to get messages from its mailbox.

The forwarding-pointer scheme has no location update cost. Because the sender delivers messages to the target agent's cached address, the agent's home workload is smaller. Even if the cache is outdated, messages can be routed to the target agent along the path. If one host on the migration path fails, however, the target agent is no longer reachable. More importantly, this protocol cannot guarantee message delivery because a message can keep chasing the target agent if the agent migrates frequently.

The FM-*-NS protocol's multihop path could degrade communication performance significantly. In contrast, an agent's mailbox migrates less frequently in JM-PL-NS and JM-PS-NS, thus shortening the message-forwarding path and reducing the communication overhead. The chasing problem is also less likely to occur in these two protocols.

Distributed-registration-based protocols

The last three protocols shown in Table 1 use synchronization to guarantee reliable message delivery. Before migrating, the mailbox informs all hosts on its migration path and waits for `ACK` messages from them. After arriving at the target host, the mailbox registers its new address with hosts on its migration path.

The sender sends messages to the cached mailbox address—say, h_k. If the mailbox has moved, h_k forwards the messages to the mailbox's current address and notifies the sender of the new address. This protocol is similar to NM-PS-SMA, but it distributes the agent home's role to all hosts on the migration path. It can also be regarded as a forwarding-pointer protocol with migration-based path compression—the agent updates all pointers on its migration path after one or several migrations.

To guarantee message delivery, the protocol must synchronize host message forwarding and mailbox migration.

If an agent migrates frequently, synchronization and location registration would make FM-*-SHM prohibitively expensive. However, if the mailbox migrates in JM mode, both the mailbox registration times and the number of hosts on its migration path decrease.

In JM-PL-SHM, the agent uses the pull operation to obtain messages from its mailbox. To guarantee message delivery, the protocol must synchronize host message forwarding and mailbox migration. In JM-PS-FS, the mailbox pushes each incoming message to its owner agent, so the protocol must also synchronize message pushing from the mailbox and the owner agent's migration. For this reason, the protocol uses full synchronization.

In a recent analysis of JM-PL-SHM,[8] we found synchronization to be quite effective. Messages are forwarded at most once to reach the receiver's mailbox, and no chasing problem exists. By properly determining the mailbox's migration frequency, designers could create a protocol that reduces both migration and delivery costs.

Our proposed scheme offers only a guideline for designing and classifying mobile agent communication protocols. The 3D model is open to further detail along each dimension to support uniform definitions of the protocol terms, assumptions, and properties. Our future work will address the design of adaptive communication protocols, moving beyond static combinations of the three design options and giving application programmers the dynamic capabilities to customize protocols at runtime. ■

Acknowledgments

We are grateful to the anonymous referees for their helpful suggestions. Our research is partially supported by the University Grant Council of Hong Kong under CERG grant B-Q518 (PolyU 5076/01E), the Hong Kong Polytechnic University under PolyU ICRG grant A-PC53, and the US National Science Foundation under grants EIA-0086260 and EIA-0115885.

References

1. D.B. Lange and M. Oshima, "Seven Good Reasons for Mobile Agents," *Comm. ACM,* vol. 42, no. 3, 1999, pp. 88-89.
2. M. Strasser, J. Baumann, and F. Hohl, "Mole—A

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on March 16, 2009 at 03:20 from IEEE Xplore. Restrictions apply.

Java Based Mobile Agent System," *Special Issues in Object-Oriented Programming: Workshop Reader of the 10th European Conf. Object-Oriented Programming* (ECOOP 96), M. Mühlhaüser, ed., Dpunkt Verlag, Heidelberg, Germany, 1996, pp. 327-334.

3. V.A. Pham and A. Karmouch, "Mobile Software Agents: An Overview," *IEEE Comm. Magazine*, vol. 36, no. 7, 1998, pp. 26-37.
4. P.T. Wojciechowski, *Algorithms for Location-Independent Communication between Mobile Agents*, tech. report DSC-2001/13, Communication Systems Dept., Swiss Federal Inst. of Technology, Zurich, Switzerland, 2001.
5. J. Cao et al., "Checkpointing and Rollback of Wide-Area Distributed Applications Using Mobile Agents," *Proc. 15th Int'l Parallel and Distributed Processing Symp.* (IPDPS 01), CD-ROM, IEEE CS Press, Los Alamitos, Calif., 2001.
6. B. Awerbuch and D. Peleg, "Online Tracking of Mobile Users," *J. ACM*, vol. 42, no. 5, 1995, pp. 1021-1058.
7. C. Perkins, ed., *IP Mobility Support*, Internet Engineering Task Force RFC2002 (Standards Track), 1996; http://www.ietf.org/rfc/rfc2002.txt.
8. X. Feng et al., "An Efficient Mailbox-Based Algorithm for Message Delivery in Mobile Agent Systems," *Proc. 5th IEEE Conf. Mobile Agents* (MA 01), IEEE CS Press, Los Alamitos, Calif., 2001, pp. 135-151.

Jiannong Cao *is an associate professor in the Department of Computing at the Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. His research interests include parallel and distributed computing, computer networks, Internet computing, mobile computing, and fault tolerance. Cao received a PhD in computer science from Washington State University. He is a member of the IEEE Computer Society, the ACM, and the American Association for the Advancement of Science. Contact him at csjcao@comp.polyu.edu.hk.*

Xinyu Feng *is a master's student in the Department of Computer Science and Technology at Nanjing University, Nanjing, China. His research focuses on networking, distributed systems, and mobile agents. Feng received a BSc in computer science from Nanjing University.*

Jian Lu *is a professor in the Department of Computer Science and Technology at Nanjing University. His research interests include Internet computing, mobile agents, componentware, and parallel object-oriented technology. Lu received a PhD in computer science from Nanjing University. He is a member of the ACM. Contact him at lj@nju.edu.cn.*

Sajal K. Das *is a professor in the Department of Computer Science and Engineering and founding director of the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas at Arlington. His research interests include resource and mobility management in wireless networks, mobile and pervasive computing, mobile Internet architectures and protocols, distributed processing, and grid computing. Das received a PhD in computer science from the University of Central Florida, Orlando. He is a member of the IEEE Computer Society, the IEEE, the ACM, the New York Academy of Sciences, and Sigma Xi. Contact him at das@cse.uta.edu.*

Computer Wants You

Computer **is always looking for interesting editorial content. In addition to our traditional articles, we have other feature sections such as Perspectives and Computing Practices as well as numerous columns to which you can contribute. Check out our author guidelines at http://www.computer.org/computer/author.htm for more information about how to contribute to your magazine.**

Innovative Technology for Computer Professionals
Computer

Authorized licensed use limited to: Hong Kong Polytechnic University. Downloaded on March 16, 2009 at 03:20 from IEEE Xplore. Restrictions apply.