PICO: A Middleware Framework for Pervasive Computing

The pervasive information community organization is a framework for creating mission-oriented dynamic communities of autonomous software entities that perform tasks for users and devices. PICO's telemedicine example scenario demonstrates its potential as a simple, unique, and versatile middleware framework for pervasive computing.

Mohan Kumar, Behrooz A. Shirazi, Sajal K. Das, Byung Y. Sung, and David Levine
University of Texas at Arlington

Mukesh Singhal
University of Kentucky

Pervasive computing can change the way we use computing devices and broaden the Internet's applications enormously. Several universities and research organizations have embarked on exciting new projects in pervasive computing. Mark Weiser predicted pervasive use of computing devices and laid the foundation for research work in this area.[1] He imagined that computing hardware and software will disappear into the background and that users will take them for granted. In the same vein, we are working on a community computing concept where users interface with services, and computing hardware and software is transparent.

The *pervasive information community organization* is a middleware framework that enhances existing Internet-based services.[2] PICO's objective is to meet the demands of time-critical applications in areas such as telemedicine, the military, and crisis management that demand automated, continual, unobtrusive services and proactive real-time collaborations among devices and software agents in dynamic, heterogeneous environments.

PICO creates mission-oriented dynamic computing communities that perform tasks for users and devices. It consists of autonomous software entities called *delegents* (or intelligent delegates) and hardware devices called *camileuns* (or connected, adaptive, mobile, intelligent, learned, efficient, ubiquitous nodes). PICO's objective is to provide "what we want, when we want, where we want, and how we want" types of services autonomously and continually. The PICO concept extends the current notion of pervasive computing—namely, that computers are everywhere. The novelty of this initiative lies in creating communities of delegents that collaborate proactively to handle dynamic information, provide selective content delivery, and facilitate application interface. In addition, delegents representing low-resource devices can carry out tasks remotely.

A case for PICO

Mahadev Satyanarayanan has identified four new areas of research—effective use of smart spaces, invisibility, localized scalability, and masking uneven conditioning.[3] In PICO, mobile and static delegents representing camileuns in communities try to use resources as effectively as possible. PICO's layered architecture attempts to mask the heterogeneity among devices and associated software. Service-provisioning communities allow a high degree of transparency between users and applications on one hand and the infrastructure on the other. Randy Katz, in his keynote address at the Per-

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/03/$17.00 © 2003 IEEE

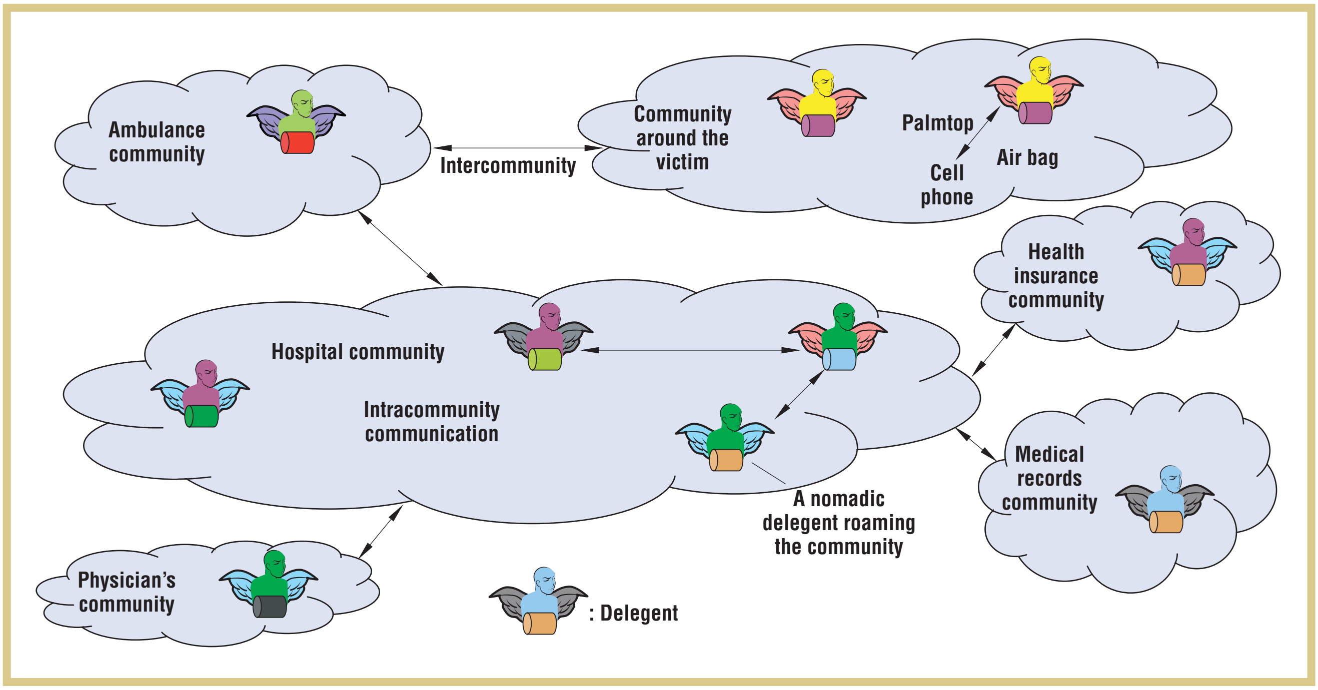


Figure 1. An example telemedicine scenario with the pervasive information community organization (color combinations depict different program modules).

vasive 2002 conference (www.cs.berkeley.edu/~randy/Talks/Pervasive2002.pdf), proposed a layered structure for services in pervasive computing, stressing the importance of network overlays, intelligence in the network, fluid software, and service provisioning. The PICO framework endeavors to fulfill the needs of the middle layers.

Currently, the trend is toward interconnecting devices and computers based on the Internet model. However, this model is lacking in these areas:

- Ability to handle dynamically changing information
- Adaptability to changing environments
- Scalability in terms of the number of users, computers, and data size

Technological developments in mobile computing and wireless communications cannot meet the demands of many critical applications requiring instant information availability. Today, we have the computing hardware and communication bandwidth to provide such services, but they are not deployed for general use, even though they have been tested in research laboratories and restricted environments. Current solutions are application-specific, device-centric, and poor in scalability. Our goal with PICO is to realize adaptability to hardware and software changes, application requirements, and the number of users and devices.

The delegents of PICO consist of different types of program modules. The two wings, roller, and head represent basic program modules. The roller and wings signify a delegent's intra- and intercommunity mobilities, respectively. The component colors and their combinations depict different program modules and delegent varieties.

We can demonstrate PICO's applicability through an example telemedicine scenario (see Figure 1). Suppose a car accident victim needs immediate attention by medical and other personnel who are in geographically distributed locations. PICO automatically creates mission-oriented communities of delegents representing doctors, hospitals, and ambulances. Then, it sets up a framework for effective collaboration to save the victim. Devices (camileuns) around the victim, such as a street camera, a cellular phone, and pocket PCs exchange sensory data. Subsequently, the delegents representing these devices create a community and contact an ambulance service. Upon arrival, the ambulance community interfaces with the hospital and other related communities to perform the required collaborative tasks in real time. Once the victim is in the care of the ambulatory emergency medical technicians, life-saving treatment can begin immediately. While an EMT documents the patient's medical abnormalities into an electronic medical record system, the ambulance community contacts the nearest and most suitable hospital and pages the appropriate on-call medical specialist automatically. The emergency room staff is alerted to the patient's impending arrival and medical condition. Vital signs are continually up-

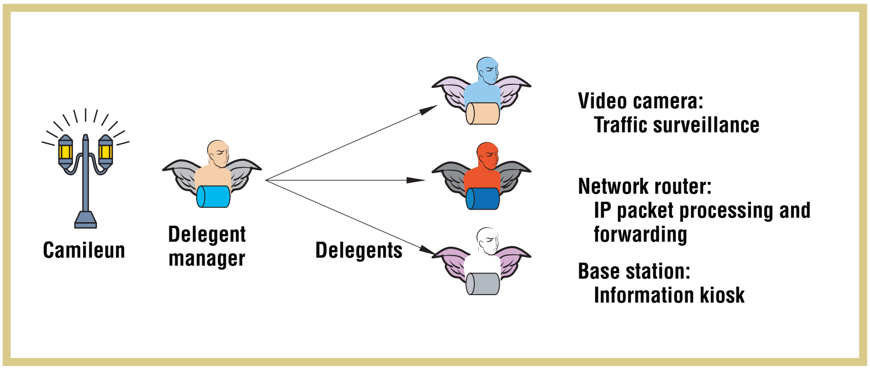


Figure 2. Example camileun and associated delegents.

dated via a continuous feed from the community around the ambulance's patient monitoring equipment—such as an EKG, pulse oximeter, and ventilator. If more emergency care is necessary, PICO communities in the hospital and ambulance (and perhaps the medical specialist's home) create a videoconferencing session to let physicians and specialists visually assess the patient and direct care through the EMT.

When the ambulance community is contacted by a community around the victim, the network elements activate the traffic community, which in turn attempts to provide a smooth and swift passage for the ambulance. Similarly, when the ambulance community contacts the hospital community, the patient's medical record is automatically preretrieved for use by the appropriate medical personnel. In Figure 1, the large community in the center belongs to a hospital and the smaller communities belong to patients and related activities. The smaller communities can seamlessly blend into the larger community (both physically and functionally).

PICO's architecture

Camileuns and delegents are PICO's basic building blocks. A delegent representing a user, application, or camileun can reside on another camileun. A community's delegents can reside on one or more camileuns. Community operations involving several delegents and the camileun resources make up the PICO architecture.

Camileuns

A camileun is a device, typically possessing one or more functionalities—such as see, hear, adapt, compute, communicate, learn, or process information. We can describe a camileun by three tuples, $C = < C_{id}, C_h, F>$, where C_{id} is the camileun identifier, C_h is the set of system characteristics, and F is the set of functionalities. For example, we describe a heart monitor camileun by $C(HR) = <\#\#\#, C_h, F>$, where ### is the identifier, C_h = <operating system, processor type, memory, I/O type; battery, wireless transceiver, ...>, and F = <ECG_monitoring, processing, communicating, ... >. Camileuns can be of different types and complexities—for example, a temperature sensing device, an active network node, or a state-of-the-art workstation. They can communicate in broadband, wired, wireless, or optical networks.

Delegents

A delegent works diligently on behalf of a camileun or user. For example, a delegent can gather information locally or remotely to collaborate with other delegents to form a computing community. A delegent is represented by the tuple $D = <D_{id}, F_d>$, where D_{id} is the delegent's identity and F_d is its functional description. Functionally, we can describe a delegent by a three tuple, $F_d = <M, R, S>$, where M is the set of program modules, S is the set of rules for delegent behavior, and L is the delegent's goal or mission. For example, a delegent associated with the streetlamp camileun, D (streetlamp, camera) can be defined as $D_1 = <M_1, R_1, S_1>$, where M_1 = <image capture, event detector, timer, ...>, R_1 = <state transition, migration, new connection,... >, and S_1 = <surveillance>. Another delegent, D (streetlamp, network router) is defined as $D_2 = <M_2, R_2, S_2>$, where M_2 = <network processor, buffer, timer, ...>, R_2 = <state transition, migration, ...>, and S_2 = <packet switching>.

A delegent receives inputs from many sources such as external events, internal events, and intentions. A delegent can be in a dormant, active, or mobile state. A dormant delegent is activated when certain events take place in its environment. A mobile delegent migrates from one camileun to another. After completing the designated tasks in the remote camileun, a delegent returns to the active state. A delegent may terminate itself after achieving its goal or mission.

Delegents + camileuns = chameleons

Delegents represent camileuns in various communities; this attribute makes camileuns adapt to different situations much like real chameleons. The streetlamp in Figure 2 is an example of a camileun. The streetlamp is equipped with a camera, a CPU, a network processor, RAM, hard disk space, and wireless transceivers. The streetlamp can function as a video camera for traffic surveillance and control, an active network element for IP packet processing and forwarding, or an information kiosk capable of exchanging information with pocket PCs carried by humans and mobile devices in passing vehicles. Typically, a camileun has a *delegent manager* responsible for coordinating with the camileun hardware as well as creating and coordinating other delegents. The DM, represented by D (streetlamp), activates three delegents to represent the streetlamp in three communities:

- Traffic surveillance and control
- Network routing
- Distributed information kiosk

For example, in the telemedicine scenario, the streetlamp is part of the traffic surveillance and the communication network communities. On sensing the accident situation, the streetlamp delegent in the traffic-surveillance community creates a smooth path for the ambulance.

PICO's layered structure

PICO is made up of four layers, as Figure 3 shows. The camileun (physical) layer consists of the computing and communication hardware, communication network, operating system, and associated drivers. The next layer comprises PICO-compliance software for the delegent-camileun interface that adapts existing hardware devices to the PICO environment. This layer also consists of communication APIs and modules as well as location- and energy-aware mechanisms. The delegent layer consists of delegents created for carrying out various tasks and services in communities. Creating, merging, and disintegrating communities takes place at the community layer. A community's functionality, mission, interest, and goals are also defined in the community layer. All layers except the lowest are also responsible for providing mobility services, resource discovery, authentication, quality of service (QoS), and resource management activities such as network monitoring, congestion control, and energy saving.

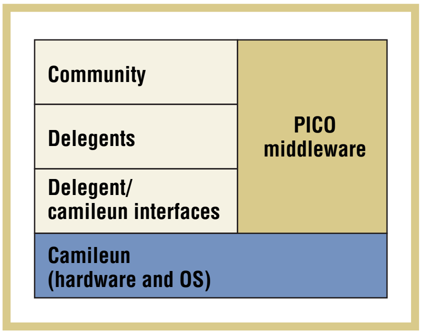


Figure 3. PICO's layered structure.

Communities

We can define a community by $P = \langle DI, GI, Ca \rangle$, where DI is the set of delegents in the community; GI is the set of community goals or missions; and Ca is the community characteristics such as community identity, number of delegents, community coordination manager, and community resources needed. As an example, we can define the community for the accident victim in Figure 1 as $P = \langle DI, GI, Ca \rangle$. Here, $DI = \langle D(\text{pocket PC}), D(\text{street camera}), D(\text{car}), D(\text{palmtop}), D(\text{cell phone}) \rangle$, where $D(x)$ is a delegent representing camileun x in this context; GI = <call emergency service, get help, …>; and Ca = <accident victim, 5, D(street camera), Internet access, telephone access, …>. A given camileun can have more than one delegent. For example, D(car) is the DM in the car's computer.

Any sequence of events can lead to the creation of communities. In Figure 1's telemedicine scenario, the accident triggers several events: D(pocket PC) detects unusual activity such as an abrupt change in beacon frequencies from other devices; D(streetlamp) detects an image of a crashed motor vehicle; and D(car) senses a sudden deceleration and halt. Each delegent individually recognizes the occurrence of an extraordinary event. One or more of these corresponding delegents attempt to contact each other, assess the situation, and form a community P, whose current goal is to call emergency services. The delegents in the community, D(cell phone) or D(streetlamp), contact an emergency service. The victim's community at the top right of Figure 1 depicts P. If the cell phone is not functioning or it does not receive acknowledgment after a certain time period, the P delegents attempt to get help via the streetlamp's connection to the traffic management network.

Pervasive devices or camileuns (such as PDAs, cell phones, watches, and computers) communicate with each other using one or more communication protocols. One or more delegents can represent each camileun. Communities of delegents provide the middleware services between the heterogeneous camileuns.

Research challenges

PICO's objective is to provide transparent, autonomous, and continual middleware services despite problems due to mobility and heterogeneity. PICO faces several research challenges, however, some of which we will investigate in the first phase.

Pervasive information acquisition and dissemination

Pervasive information acquisition and processing is an important issue in the PICO project. Delegents acquire information within their community and in remote communities. $D(x)$ can reside in camileun y while gathering information to ensure optimal use of resources. Methods for acquiring and disseminating information include

- Creating communities
- Deploying context-aware services
- Facilitating delegent migration
- Caching, prefetching, and push-caching policies

Database systems provide fast and reliable information storage and retrieval, which is crucial in PICO. A database system for dynamic wireless systems must incorporate mobility in the database system because positions must be continuously updated. In wireless computing environments, frequent position updates would impose serious performance and wireless-bandwidth overhead. In the future, we plan to develop methods for capturing and maintaining system state on databases that are distributed among the moving hosts. We are also investigating dynamic attributes that represent the position of moving objects. Delegents collect the system's global state and store it in a distributed database.

Context- and location-aware computing in wireless mobile environments

Rapid advances in a range of wireless access technologies along with an industry-wide IP convergence have set the stage for context-aware computing. By using various technologies to infer the user's activity state (for example, is the user walking or driving, at the office, at home, or in a public environment?), future networks can intelligently manage the content and mode of information delivery.

The user's location information plays an important role in defining this context. We are exploring current trends toward integrated access technologies in wireless mobile environments and generic functions such as location update and paging, which are an important part of most location management systems.[4] Symbolic location information for such an abstract location management architecture is useful for profiling users' personal mobility patterns. With this data, we can build a predictive framework that supports numerous location-aware services. Our goal with PICO is to design a community-computing-based architecture to provide location support in real-life applications. This architecture will be based on the definition of open interfaces using the industry-standard Lightweight Directory Access Protocol[5] and will provide a flexible way to exchange and coordinate location information across multiple systems.

Service-provider communities

A PICO service-provider community has a predetermined mission, and its delegents carry out prespecified tasks. This is in contrast to communities created dynamically when events occur in order to achieve a specific goal.

To illustrate this concept, consider the example of a network service-provider community P_n that consists of delegents distributed across the network. In the telemedicine scenario, we must set up a reliable communication channel between the victim's site and the hospital ER to let doctors interact and direct appropriate care. The hospital community contacts a network service-provider community such as P_n to establish the reliable communication channel. P_n is a distributed service-provider community with delegents at strategic locations to monitor and manage the network. P_n recognizes a critical situation and reorganizes the network to ensure a secure, reliable channel for the telemedicine application.

Security mechanisms

Services and information available on and exchanged between nodes (such as delegents) are vulnerable to security and privacy breaches that standard security approaches are not prepared to deal with. Mobile code executes on untrusted hosts and, hence, is subject to corruption, as are the objects and environments they manipulate. Existing approaches[6] make certain assumptions that might not hold in the PICO environment causing major hindrances in effectively using communication and shared information. We plan to investigate the applicability of tools such as security mediators and virtual firewalls[6] to secure information in the PICO environment. Particularly, we are investigating the development of static and dynamic communities that can provide security services.

QoS architecture development

Guaranteeing sustained QoS in dynamic situations requires accurate system awareness (monitoring) and proactive QoS and resource management. Integrated- and differentiated-service-based mechanisms fail to provide guaranteed QoS, although the latter are scalable. In PICO, delegents and their

Related Work

Recently, pervasive computing has been an active area of research attracting substantial attention by several research groups in universities as well as industries. Here is a list of URLs to visit for up-to-date information on these projects:

Aura project at the Carnegie Mellon University:
www-2.cs.cmu.edu/~aura
Oxygen Project at the Massachusetts Institute of Technology:
http://oxygen.lcs.mit.edu
Endeavour Project at the University of Berkeley:
http://endeavour.cs.berkeley.edu
Contextual Computing at Georgia Tech:
www.gvu.gatech.edu/ccg
Sentient Computing at Cambridge University:
www.uk.research.att.com/spirit
Portolano Project at the University of Washington:
http://portolano.cs.washington.edu
PIMA Project at IBM:
www.research.ibm.com/PIMA

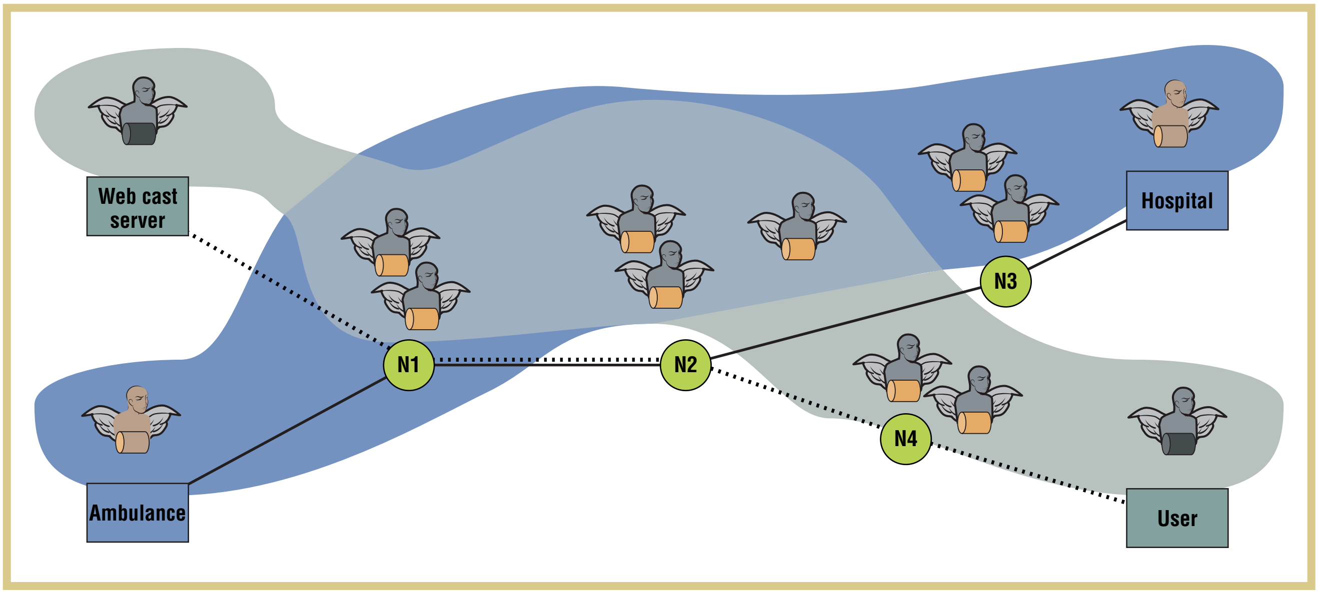


Figure 4. Quality of service adaptation.

communities can adapt to changing QoS policies swiftly and transparently by facilitating better QoS management. A PICO-compliant network with service-provider communities can ensure guaranteed QoS for critical (or high-priority) communities.

QoS provisioning and resource management are critical to PICO's applications such as telemedicine and crisis management. Such dynamic situations demand accurate system awareness (through constant monitoring) and proactive management of QoS and resources. Service-provider communities are made up of middleware delegents that reside and roam within the network, possess current knowledge of network state, and make proactive decisions to ensure QoS provisioning. Resource manager delegents discover and allocate resources. PICO-based middleware support for QoS and resource management can be provided with the help of profiling, monitoring, and dynamic adaptation.

QoS profiles include application description (type and priority, parameters, and levels), resource description (a set of participating components, OS, and communication channels), and adaptation policies (when and how to adapt to changing environments). Resource manager, profiler, application, and monitoring delegents collaborate with each other to build application and resource profiles. In the telemedicine scenario, for example, when it is necessary to stream video from the ambulance to the hospital, the resource manager delegents know the transmission requirements and resource availability, thus saving precious time. Profiler delegents constantly update application profiles by continuous communication with application delegents. This profiling also facilitates just-in-time QoS adaptations. Likewise, monitoring delegents collect network-related parameters and update network profiles. As these delegents reside in communities, they share information with other delegents to ensure QoS enforcement.

For timely QoS adaptations, modified just-in-time collaboration among the communities can be established. The tasks carried out by service-provider communities are

- Continuous monitoring and profiling to ensure early prediction and detection of possible QoS violations
- System diagnosis to identify the causes of violations and determine possible actions
- Analysis of different solutions to perform QoS management and select the best options

Figure 4 illustrates an example network that helps describe the QoS management process. An ongoing service exists between the Web casting server and the user's terminal via network nodes N1, N2 and N3. This communication path has a delegent community that enforces QoS. Once an accident takes place, the PICO system must form communities in the manner we described earlier for the telemedicine scenario.

The communities at the ambulance and hospital need a channel to transmit a video stream of the accident victim from the ambulance location. Such a requirement necessitates just-in-time availability of a jitter-free, secure, and reliable communication channel with adequate bandwidth. The best routing path for this streaming involves nodes N1 and N3. The service-provider community negotiates with the community responsible for the sports Web cast to provide adequate bandwidth for the emergency transmission's communication. Because the application profiles (for the sports and telemedicine events) contain information about application types and priorities, a quick decision is made by delegents at nodes N1 and N3 to reallocate bandwidth so as to let packets flow between the ambulance and hospital. The community responsible for the sports event can use caching, compression, graceful degradation, and replication mechanisms to deliver a combination of up-to date scores, statistics, and replays (and even cached advertisements) at N4 transparently to the user. Because delegents and their communities are software entities, they adapt to changing QoS policies swiftly and transparently to facilitate just-in-time services. Because the communication about the accident victim is critical (defined in the application profile), the resource-manager delegent interacts proactively with monitor delegents and searches for an alternative path to ensure fault tolerance. Furthermore, the delegents at N4 may reassign the sports Web cast to another available path.

> Because delegents and their communities are software entities, they adapt to changing QoS policies swiftly and transparently to facilitate just-in-time services.

QoS adaptations in this scenario are possible due to the just-in-time communication, collaboration, and community-based information sharing among delegents. We plan to measure performance metrics such as the QoS violation rate, ratio of QoS violations to resources used, and average distance from violations.

Ongoing work

We have made significant progress in the research and developmental work on several fronts. For example, we have developed formal definitions and models for delegents, delegent-to-delegent communications, service discovery, and service-provider communities (http://cse.uta.edu/pico@cse). We are extending these service-provider communities to develop ambulance, hospital, medical, physician, and other related records. Also, we have developed several generalized service-provider communities that can be used in the telemedicine scenario and other applications.

Delegents and communities

We have defined special-purpose delegents for carrying out missions such as communication, service discovery, ECG monitoring, user profiling, and device management. Also, we have developed prototype models to depict general-purpose service-provider communities for simple applications such as printing.

We have completed the design, implementation, and evaluation of a communication framework for delegents in service-provider communities. Software agents communicate using direct or indirect communication. Most multiagent systems support one of these methods, resulting in overhead for certain types of shared documents. In a community, document size and type varies depending on the type of events and community goals, so it is inefficient to use any one method. To overcome this problem, we developed a communication framework that adapts the best delegent-to-delegent communication method based on the situation as well as document size and type in various service-provider communities.

Service-provider communities

We are currently developing service-provider communities for managing communication networks (traditional and mobile) to enhance TCP/IP and mobile IP performances. This research work assumes the existence of active elements in the communication network. Delegents and their communities execute on active elements to carry out specific goals and missions.

A service-discovery mechanism that exploits the community computing concept is also under development. Delegents represent distributed directories, servers, clients, and services. Communities of delegents representing servers and devices that host services and client applications, respectively, are formed to provide anytime, anywhere service discovery in mobile and heterogeneous environments. The community-based mechanism attempts to provide responsive and effective service discovery that lets clients specify what to look for rather than where to look for it.

We could deploy PICO-driven environments for applications in almost all aspects of life, including military, education, office management, banking, real estate, entertainment, crisis management, and disaster prevention. For example, in a PICO-enabled office management environment, a CEO can set up a meeting by simply specifying the meeting time. The CEO's delegents will automatically form a community with the delegents of all involved parties to schedule the meeting. The delegents in the meeting community use the participants' calendar information to resolve conflicts and determine the mode of participation by different employees.

As far as we know, PICO differs from any other existing project. We are in the process developing formalisms for creating delegents and communities and their behavior. We have made significant progress in developing service-provider communities, delegent structures and communications, and service discovery. This is an ongoing project with wide implications. We are also applying the community computing concept in two new projects in the areas of manufacturing and pervasively secure environments.

ACKNOWLEDGMENTS

We acknowledge the technical comments and feedback received from Farhad Kamangar and Gergely Zaruba of the University of Texas at Arlington. Graduate students of the PICO lab at the Department of Computer Science and Engineering at University of Texas at Arlington have also contributed to the ongoing research work. R.A. Peshock, Alice Marcee, and Michael Chwialkowski of the UT Southwestern Medical Center in Dallas have been helpful in giving us valuable inputs about the telemedicine issues. We are also grateful to the anonymous referees for their constructive comments. This material is based on work supported by the National Science Foundation under Grant No. 0129682.

REFERENCES

1. M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, no. 3, Sept. 1991, pp. 66–75.

2. M. Kumar et al., "Pervasive Information Community Organization (PICO) of Camileuns and Delegents for Future Global Networking," *Proc. LSN Workshop*, 2001; www.ngi-supernet.org/lsn2000/UT-Arlington.pdf.

3. M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Computing*, Aug. 2001, pp. 10–17.

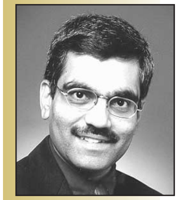
4. A. Bhattacharya and S.K. Das, "LeZi-update: An Information-Theoretic Approach to Track Mobile Users in PCS Networks," *Proc. Ann. Int'l Conf. Mobile Computing and Networking*, ACM Press, 1999, pp. 1–12.

5. M. Wahl et al., "Lightweight Directory Access Protocol (v3)," RFC 2251 (Draft standard), Internet Eng. Task Force, Dec. 1997; www.ietf.org/rfc/rfc2251.txt.

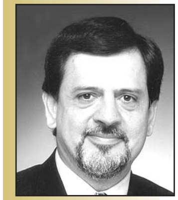
6. W.M. Farmer, J.D. Guttman, and V. Swarup, "Security for Mobile Agents, Issues and Requirements," *Proc. Nat'l Information Systems Security Conf.* (NISSC'96), Nat'l Inst. Standards and Technology, 1996, pp. 591–597.

For more information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

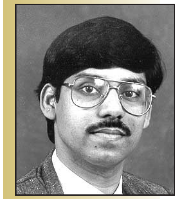
the AUTHORS



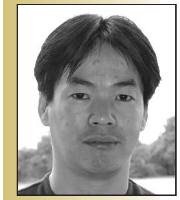
Mohan Kumar is an associate professor in the Department of Computer Science and Engineering at the University of Texas at Arlington. His research areas include mobile computing, wireless networks, pervasive computing, and parallel and distributed computing. He has a PhD in computer science from the Indian Institute of Science. He is a senior member of the IEEE. Contact him at the Dept. of Computer Science and Eng., PO Box 19015, Arlington, TX 76019; kumar@cse.uta.edu.



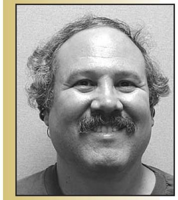
Behrooz A. Shirazi is a professor and the chair of the Department of Computer Science and Engineering at University of Texas at Arlington. He has conducted research in pervasive computing, software tools, distributed real-time systems, scheduling and load balancing, and parallel and distributed systems. He has a PhD in computer science from the University of Oklahoma. Contact him at the Dept. of Computer Science and Eng., PO Box 19015, Arlington, TX 76019; shirazi@cse.uta.edu.



Sajal K. Das is a professor of computer science and engineering and is the 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, wireless multimedia and QoS provisioning, sensor networks, mobile Internet architectures and protocols, distributed processing, and grid computing. He has a PhD in computer science from the University of Central Florida, Orlando. Contact him at the Dept. of Computer Science and Eng., PO Box 23015, Arlington, TX 76019; das@cse.uta.edu.



Byung Y. Sung is a PhD candidate in the Department of Computer Science and Engineering at the University of Texas at Arlington. His research interests include pervasive and ubiquitous computing, parallel and distributed systems, and mobile computing. He has an MS in computer science and engineering from the University of Texas at Arlington. He is an IEEE member. Contact him at the Dept. of Computer Science and Eng., PO Box 19015, Arlington, TX 76010; sung@cse.uta.edu.



David Levine is a visiting assistant professor in the Department of Computer Science and Engineering at the University of Texas at Arlington. His interests include grid and cluster computing, software mobile objects, software agent communication and security, and ubiquitous computer environments, particularly applied to the needs of people with disabilities. He has an MS from the University of Texas at Austin. Contact him at the Dept. of Computer Science and Eng., PO Box 23015, Arlington, TX 76019; levine@cse.uta.edu.

Mukesh Singhal is a professor and the Gartener Group Endowed Chair in Network Engineering in the Department of Computer Science at the University of Kentucky, Lexington. His research interests include distributed systems, mobile computing, computer networks, computer security, and performance evaluation. He has a PhD in computer science from the University of Maryland, College Park. Contact him at the Dept. of Computer Science, University of Kentucky, Lexington, KY 40506; singhal@cs.uky.edu.