♦ Advanced Services: Changing How We Communicate

Raymond O. Colbert, Diane S. Compton, Randy L. Hackbarth, James D. Herbsleb, Laurie A. Hoadley, and Graham J. Wills

This paper introduces three next-generation collaboration services designed to restore the human social interactions often lost in geographically distributed work environments. These services—TeamPortal™, ConnectIcon™, and OpenChannel™—will be delivered as a trial and demonstration system on the service access, mediation, and management (SAMM) platform. SAMM technology allows service providers to use the standard Lucent Softswitch to add converged voice and data services to their networks by employing service frameworks. These frameworks handle service interactions, including service session management; operations, administration, maintenance, and provisioning (OAM&P); and billing. As examples of new revenue-generating services, the collaboration services demonstrate the advantages of a packet-based core network. This type of network provides always-on communication that creates an awareness of the presence of other team members and enables instant meetings and World-Wide-Web-based meeting areas for geographically distributed teams. SAMM, which interfaces with the Lucent Softswitch through the Java Telephony Application Programming Interface (JTAPI), takes advantage of all available means of establishing communication, including "plain old telephone service" (POTS), text-based messaging, and voice over IP (VoIP) using softphones, session initiation protocol (SIP), and ITU-T H.323 endpoints.*

Introduction

The convergence of voice and data traffic on Internet protocol (IP) networks provides opportunities to create radically new services that will impact the core network of the future.[1] We describe a suite of services that builds on newly dominant World-Wide-Web-based communication (such as instant messaging, chat rooms, and directory services) and on traditional telephony (such as conference bridge and call forwarding) to create new, hybrid forms of communication.

It has long been claimed that the Internet would make distance irrelevant, yet many businesses struggle as economic forces conspire to make them more globally dispersed. As our research on geographically distributed work within Lucent has shown, work distributed across sites takes as much as two or three times longer to accomplish than comparable work performed at a single site.[2] The primary problems are finding the right person, contacting that person, and having effective collaborative sessions. Keeping priorities, plans, and work activities aligned across sites is also difficult, because people are removed from the day-to-day thinking and events in sites geographically remote from one another. Ironically, as companies increase their competitive advantage by building international and regional relationships and sites, communication delays that result from distributed work sites may decrease their speed and efficiency.

To counteract this effect, companies have been exploring and implementing products like instant messaging, Lotus Notes,* and NetMeeting* conferencing software. However, these data tools are not integrated with the core communication technologies, which limits their effectiveness and availability. Installation and use of these tools require investigation and action by the end user. A much more effective method is to integrate collaboration services with existing communication services, such as unified messaging, Web portals, and phones. Integration reduces the learning curve and limits the amount of change experienced in the everyday activity of the user.

Research conducted on the work environment at Lucent provided the following insights:

- People must be able to talk about and resolve issues as soon as possible after they arise. Otherwise, not only is there delay, but people also continue working, only to find that they have to redo the work once the issue is resolved.
- People must also have ready access to the type of background information exchanged in hallways, over lunch, and around the water cooler. From a distance it is not always clear what information one's colleagues will need, nor is it always obvious when one needs to ask a question. Frequent informal conversations are essential for keeping work synchronized and for establishing effective relationships.
- Finding information is important, but often it is even more important to find the right people. The things that slow people down are not usually just the lack of information, but the lack of someone to help interpret the information and advise them about how to resolve key issues.

What does all this mean to service providers? In addition to handling increased traffic, communication networks will need to expose additional data. For example, user activity and location data will be needed to support services that run on the network. The networks must also offer more ways to integrate different

Panel 1. Abbreviations, Acronyms, and Terms

API—application programming interface
applet—a small Java* application that can be run in a Web browser
E1—European primary signal rate that supports 32 64-kb/s (2.048 Mb/s) data or digitized voice channels
H.323—ITU standard for multimedia (real-time voice, data, video) communication across IP-based networks
HTTP—hypertext transport protocol
ID—identification
IETF—Internet Engineering Task Force
IN—intelligent network
IP—Internet protocol
ITU—International Telecommunication Union
ITU-T—International Telecommunication Union, Telecommunication Standardization Sector
JAIN*—Java* APIs for Integrated Networks
JTAPI—Java* Telephony API
LDAP—lightweight directory access protocol
NAR—North American region
OAM&P—operations, administration, maintenance, and provisioning
Parlay—an open API specified by the Parlay Group to enable secure public access to core capabilities of telecommunication and data networks
PC—personal computer
PDA—personal digital assistant
POP—point of presence
POTS—"plain old telephone service"
PSTN—public switched telephone network
SAMM—service access, mediation, and management
SCP—service control point
SIP—session initiation protocol
SS7—Signaling System 7
T1—terrestrial (North American) facility to transport 24 64-kb/s data or digitized voice channels at primary rate of 1.544 Mb/s
TCAP—transaction capability application part
URL—uniform resource locator
VoIP—voice over IP
VPN—virtual private network
XML—Extensible Markup Language

modes of communication and to provide an extended service logic that can help users navigate sensibly through this greatly expanded space of possibilities.

This paper introduces three next-generation collaboration services designed to restore the social interactions lost in a distributed work environment. Developed by Bell Labs researchers, these services include the following:

- *TeamPortal™ service*. This Web-based portal provides unified calendars and presence information about team members and can contact them in several ways.
- *ConnectIcon™ service*. This service allows a user to send a visual invitation to communicate. It provides the receiver with the sender's presence information and a variety of means for establishing contact.
- *OpenChannel™ service*. This service provides a Web-based meeting area for geographically distributed teams, identifies speakers, and moves among several levels of participation.

The remaining sections of this paper describe the deployment of these collaboration-based converged services on the service access, mediation, and management (SAMM) platform. The section immediately below, "Background," provides a brief overview of the Lucent packet switching solutions for service providers and the role of SAMM in these solutions. The section following it, "Collaboration Services," presents the services currently deployed using SAMM. In the two subsequent sections, "Architecture Overview" outlines the SAMM architecture and "SAMM in the Network" describes how SAMM connects to the core public switched telephone network (PSTN). The final section, "Service Scenarios," presents user scenarios and the resulting call flows of the collaboration services.

Background

The need for always-on communication and the increased competition at the service provider level have stressed traditional circuit-based core networks. For the first time, service providers are exploring wide-scale conversion to packet-based switching.

Why is packet switching creating so much excitement? The answer is simple: Packet technology provides freedom from a dedicated call path. Instead of building a complete circuit connection that must remain active for the duration of the call, service providers only have to balance the bursts of data and voice created when information is actually being sent through the network. Eliminating the dedicated call path is what enables the economical implementation of next-generation services, many of which rely on long-term connectivity. OpenChannel™ is one example of such a service.

The Lucent packet switching solutions, including SAMM, eliminate the need for a dedicated call path, allowing service providers to increase network efficiency and to cost effectively deliver next-generation data and voice services to their customers. SAMM enhances the Lucent Softswitch[3] by providing frameworks that ensure the features developed by the service provider and other third parties using the Full Circle™ application programming interface (API)[4] interact appropriately with existing voice and data services. The initial services created with this environment include the collaboration services described in this paper.

The services discussed in this paper will be delivered as part of a trial and demonstration system. This trial demonstrates technology that allows service providers to introduce advanced collaboration services to one of their key business customers. By introducing next-generation services into their network, service providers will gain real-life knowledge and experience with packet technology before they begin a large-scale network conversion effort. This type of knowledge will give service providers a better understanding of the challenges and benefits of packet conversion and advanced services.

The next section discusses the collaboration services available through the trial and demonstration system.

Collaboration Services

We have designed and built three new collaboration services—TeamPortal™, ConnectIcon™, and OpenChannel™—that take advantage of SAMM programmability to build the kinds of services that address the real problems of distributed work. In this section we briefly describe them from a user's point of view.

The services shown here incorporate elements that are seen in research prototypes, as well as a few elements available in isolated commercial tools. For example, basic presence information about who logged on to an instant messaging server has existed in such services as the AOL Instant Messenger* for some time, and similar functionality is finding its way into the workplace.[5] Shared online calendars are relatively common, and click-to-dial is now available in a number of voice-over-IP (VoIP) services.

However, the services described here are unique in several respects. First, they extend the idea of a user's presence from simply being logged on to a much richer set of information garnered from multiple devices. Second, they integrate useful functions with linked views that provide interoperation. Finally, the ConnectIcon and OpenChannel services represent completely new paradigms for communication. They are neither exclusively synchronous nor asynchronous, but rather they help teams move seamlessly between these two fundamental modes of collaboration.

TeamPortal Service

In globally distributed work, it is very difficult to know when your co-workers are available. They may live in many different time zones and observe entirely different holidays. It can be very frustrating, after calling your colleague in Bangalore all day, to discover that it is Independence Day in India. In addition, it would be useful to know the current state of your colleagues' devices—that is, if they have recently used their computer keyboards or their office phones. Placing calls to the best device at any given moment with only a click makes voice contact far faster and more likely to be successfully completed.

TeamPortal service, shown in **Figure 1**, allows a user to select—from a company directory or a local address book—a set of people with whom he or she needs to stay in close contact. For this set of people, TeamPortal displays the correct set of world clocks, as well as world calendars that list holidays for the relevant countries. With a single click, the user can also access permitted entries on each person's individual calendar.

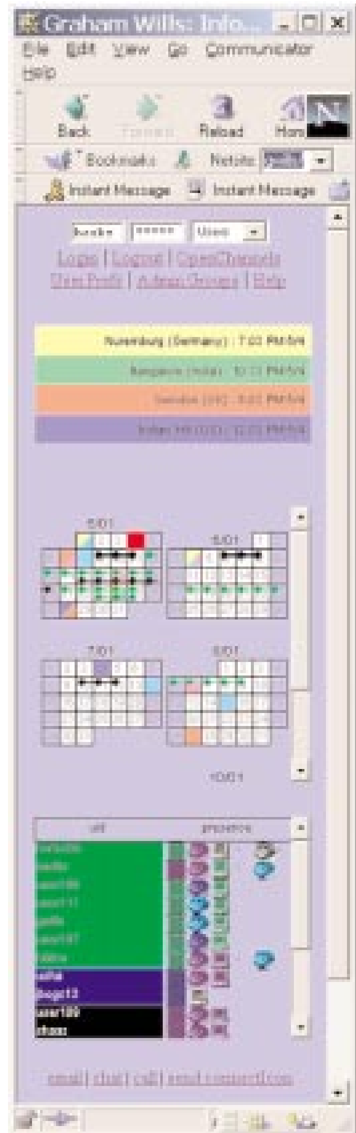


Figure 1.
TeamPortal™ service.

In the presence panel, the service begins to take advantage of convergence. A set of icons to the right of each person's name indicates the recent activity (or inactivity) of his or her devices. In this example, we see that herbsleb last used his office phone (represented by the leftmost phone icon) more than a day ago and is currently using TeamPortal (the door icon is green). User gwills, whose TeamPortal we are viewing, and user hibino are also currently using TeamPortal.

User hibino has registered both her PC and a second phone; she has not used her PC or her first phone in more than a week. The devices are divided into groups corresponding to different locations. In this example, only herbsleb, gwills, and hibino have devices registered at more than one location.

If a user wants to contact herbsleb, he or she can simply click on herbsleb's user ID and TeamPortal will initiate a phone call between the user's default phone and the one herbsleb specified as his default. Depending on the users' preferences and the type of phone clicked on, TeamPortal will connect the two parties using softphones or "plain old telephone service" (POTS) phones.

User hibino has two phones to which TeamPortal could route calls. She used a phone at one of her locations yesterday and is currently logged on to TeamPortal. It makes sense to try the phone more recently used, rather than the rarely used one at the other location. However, TeamPortal does not insist on that choice; the user can decide which phone to call by clicking on the phone icon.

The TeamPortal service also supports these capabilities beyond the user's current team or set of close contacts. Suppose the user needs to contact someone on a particular product's marketing team. When the user navigates to the team's Web page, he or she sees the team's TeamPortal, consisting of the correct world clocks, calendar, and presence information for the team. If the team has designated a contact person or a policy for specifying a contact person, only that person's contact and presence information are displayed.

ConnectIcon Service

Work distributed among numerous locations takes several times as long to complete as comparable same-site work. How do you get the right people together at the same time to resolve an issue, to negotiate a solution, or to pool their knowledge? You would like to get together as soon as all the essential people are prepared and available. ConnectIcon service allows communication that closely approximates this ideal.

The sender configures it with a sentence or two, describing the reason for the invitation, the identity of the receivers, and the uniform resource locators (URLs) for any relevant materials. If the receiver also has TeamPortal service, the ConnectIcon is displayed in TeamPortal, as shown in **Figure 2**. Placing the mouse over the ConnectIcon displays a text message, and a right click generates a pop-up menu that allows the receiver to access any URLs specified by the sender. The receiver can also access the sender's calendar and initiate a chat, e-mail, or audio connection. In addition, the ConnectIcon also displays presence data, allowing the receiver to see if it is a good time to initiate contact and take care of the task. As soon as the receiver is prepared and the sender is available, contact can be initiated.

For those without TeamPortal service, the ConnectIcon is delivered as a URL attachment to an e-mail message. Clicking the URL brings up a small, iconified picture of the sender and provides the same functionality as described above.

OpenChannel Service

The ability of co-located colleagues working together to stay closely in touch just by talking to each other is a huge advantage, particularly in stressful or crisis situations. In fact, some organizations have experimented with locating everyone in a single room—a "war room"—as a way of speeding up difficult tasks requiring coordinated activity.

Establishing long-standing audio connections can create a similar and, in some ways, improved functionality. Anyone needed, anywhere in the world, can participate in ongoing work. Talking provides a forum for people to ask questions and to describe new findings. This ability can accelerate the resolution of many problems, by passing problem solving off among team members in widely separated time zones. OpenChannel service provides the means for team members to coordinate their work and share current status during whatever overlap in working hours is available.

Placing all participants in an application-sharing session enhances the basic long-standing audio connection. They can jointly view and edit documents or whatever other applications might form the basis of collaborative activity. In addition, the OpenChannel service gives participants the opportunity to turn off the sound if it becomes bothersome and to monitor the audio channel visually, as shown in **Figure 3**. The visualization shows who is participating in the

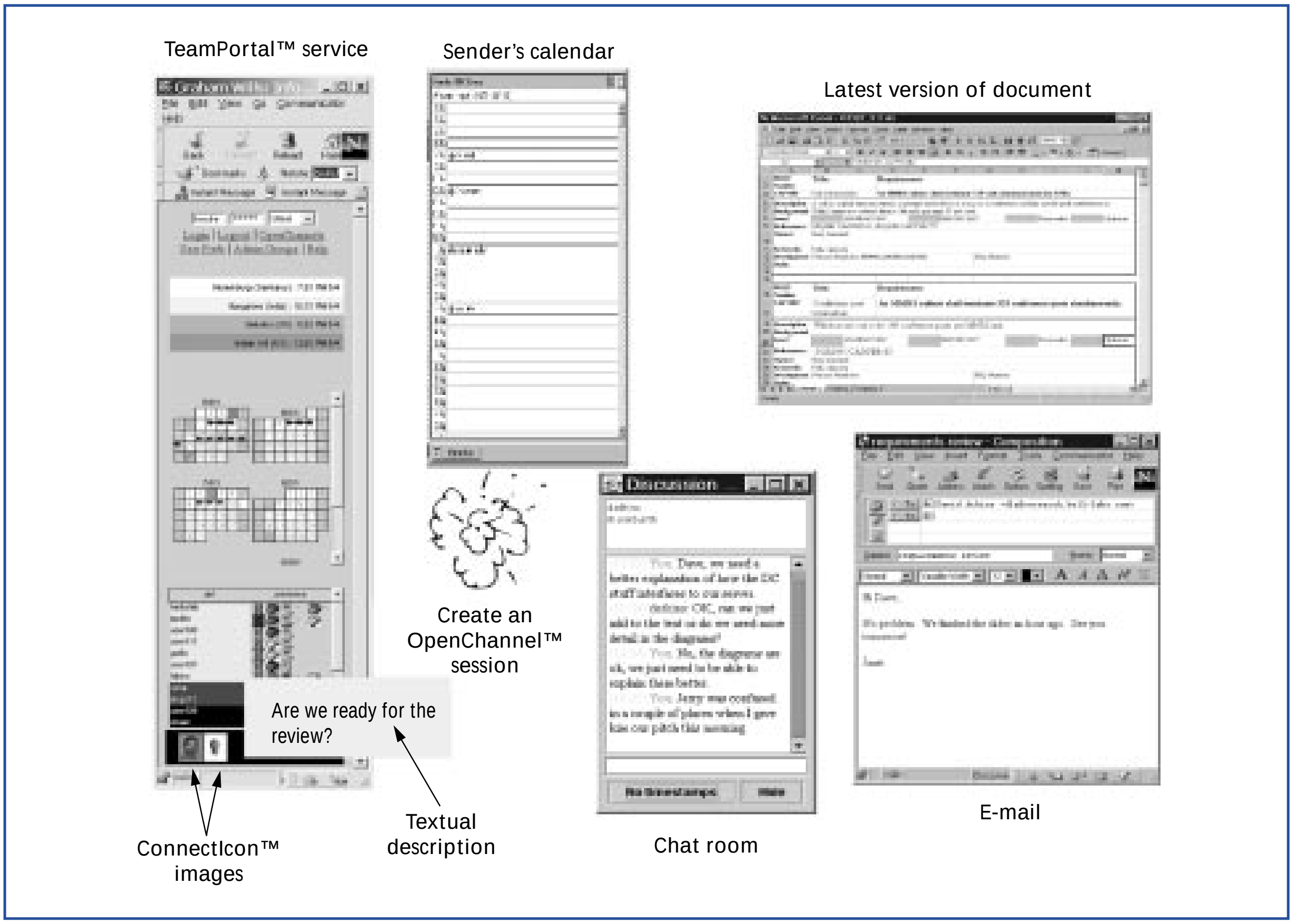


Figure 2.
ConnectIcon™ service.

OpenChannel service, who has spoken recently, and what applications are open in the application-sharing session. A user can listen to and/or monitor several channels at once.

Finally, OpenChannel service provides recording and replay capabilities. The entire session—including application-sharing activity, with pointing, scrolling, and editing—can be captured and replayed at original or accelerated speed. In addition, a participant monitoring visually may decide, on the basis of who is speaking and/or what documents they are sharing, to replay the last few minutes to see if the topic is of interest. If it is, he or she can decide whether to join more actively.

Together, these three services offer truly effective ways of finding people, connecting with them as soon as it is possible to fruitfully do so, and providing a channel for extended background and foreground communication. By taking full advantage of the convergence offered by SAMM, Lucent is providing radically new capabilities that increase the demand for its customers' services.

Architecture Overview

Collaboration services are implemented using application and media servers supported through SAMM. Based on the Lucent Softswitch, SAMM is a platform for programmable converged voice and data services in packet networks.[1] It provides a service logic execution environment for service sessions,[6] multimedia voice/video/data services, converged telephony/Internet services, enhancements to the Full Circle API, and rapid service authoring and deployment by service provider or third parties. The collaborative services are provided to intelligent soft

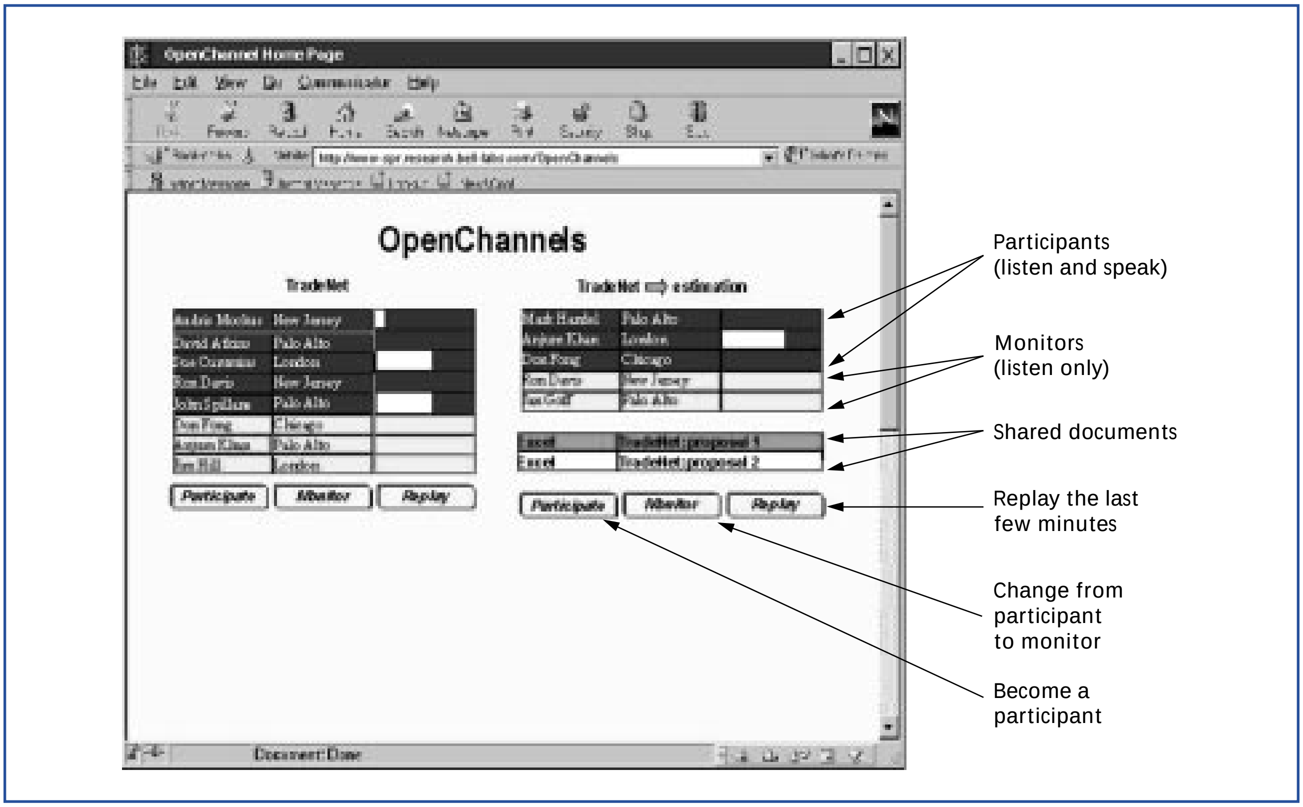


Figure 3.
OpenChannel™ service.

endpoints using multimedia-enabled protocols, such as session initiation protocol (SIP)[7] and ITU-T Recommendation H.323,[8] and a Web server interface.

SAMM incorporates a services and operations, administration, maintenance, and provisioning (OAM&P) framework on the Lucent Softswitch base. The framework enables the blending and deployment of services in a generic, extensible fashion. Together, the canonical call model, device servers, and the service framework allow the core service logic to be access and transport neutral and to interwork calls among clients communicating with diverse protocols.

To support next-generation services, the services framework, depicted in **Figure 4**, enables SAMM to provide the following three basic types of critical functionality:

- *Multimedia and converged voice/data handling*. The *access session*, which provides click-to-activate service, enables comprehensive access to next-generation services. The *service session*, which provides a context for multimedia, multi-resource activities, enables sessions beyond the conventional voice call and receives both call-related events and multimedia non-call-related events (potentially received from sources such as e-mail servers, Web servers, and SIP proxy servers).
- *Multiple applications/services capability*. As opposed to being designed for a single application at a time, SAMM is intended to support multiple applications/services for service providers to offer to subscribers. Interactions between the multiple applications/services and their components must be managed based on service deployment plans of service providers and on customer preferences. The *service logic component bus* is software that manages these interactions.
- *OAM&P designed for multiple applications and programmability*. Provisioning, measurements, billing, and policy management are variously needed for new applications/services. The

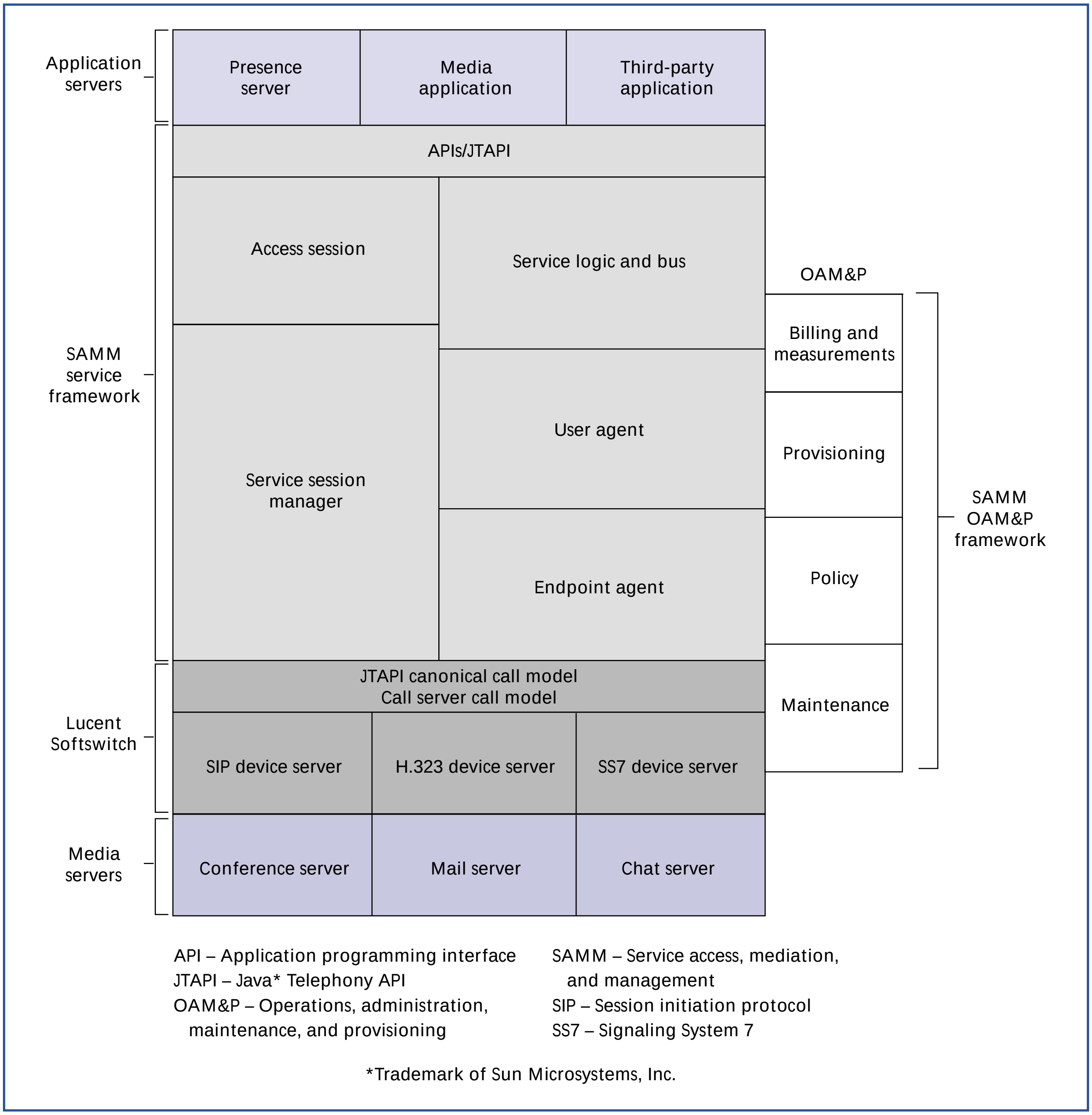


Figure 4.
Service framework components.

OAM&P framework provides a programmable structure that has uniform software connectivity to provisioning databases, billing systems, and policy engines.

The principal network components that enable converged collaboration services, as shown in Figure 4, include:

- *Application server components*, such as the presence server, media application, and third-party applications;
- *Service framework components*, comprising the access session, service session manager, service logic, user agent, and endpoint agent;

- *Lucent SoftSwitch components*, comprising protocol device servers; and
- *Media servers' components*, including the conference server, mail server, and chat server.

In addition to the network components, other components residing on soft endpoints on the customer premises are used to enable the converged collaboration services to work with the network. They include:

- The *presence client*, which consists of a presence activity application running on a user's soft endpoint and a set of collaboratory services applets running in a browser on the user's soft endpoint; and
- Additional *soft endpoint components*, including an administrative user interface client, softphone clients (such as the SIP or H.323 clients), and a media server client.

Typically, the soft endpoint is a PC, but it could be any communication device, such as a personal digital assistant (PDA) or a wireless phone that supports a browser.

Service Framework

The service framework is the architectural model that supports blending and deployment of services that can be extended as needed in a generic fashion. The service framework architecture leverages object-oriented mechanisms to maximize independence in deploying new services. The components of the service framework include:

- The *access session*, which provides a point of contact between the intelligent endpoint and the service provider. It provides a Web browser interface to the soft endpoints of the end user, performs authentication, and initiates the service chosen by the end user, based on his or her personalized preferences. The access session comprises a Web server and a standalone Java* application.
- The *service session manager*, implemented as a Java application, contains the following components:
  – The *service session component*, which manages the multimedia/multiparty context for a user's "calls," to facilitate intelligent service actions on behalf of the end user. The service session provides service blending by integrating input from the core service logic, user agents, and endpoint agents to customize service requests.
  – The core *service logic components*, service drivers that are independent of access type, transport type, and transport protocol. They receive events from the service session. The *service logic component bus*[6] provides service dispatching for each event.
  – The *user agent component*, which customizes services for a specific user through subscriber profile data. It interfaces to service sessions and core service logic.
  – The *endpoint agent component*, which represents the capabilities of the terminal. It interfaces with the service session. The service session sends the endpoint agent generic requests, such as "Alert Subscriber." The endpoint agent then customizes call treatment based on the endpoints' media capabilities, changing "Alert Subscriber" to "Do a Screen POP to a softphone," or "Ring an analog line." The service session sends the customized treatment to the appropriate device server via a Java Telephone API (JTAPI).[9]

The service framework interfaces with the Lucent SoftSwitch using JTAPI. Components are classes, which inherit from SAMM abstract classes; they plug into the service framework and into the OAM&P framework. The APIs supported by SAMM reflect the service framework model. Advanced services are provided through the service framework. The components leverage polymorphism, which means they communicate with each other based on a general interface to a general class. The specific subclass of the particular component instance customizes behavior.

The collaboration services "plug in" to this service framework, where service interactions are resolved. To customize the component classes, the collaboration services have been designed to implement not only these classes, but also subclasses.

Presence Server and Clients

The heart of the collaboration services is the *presence server*, which is responsible for maintaining presence data for the registered clients. It maintains lists of subscribers and notifies subscribers of changes in status. The presence server contains a lightweight directory access protocol (LDAP)[10] database in which presence information is maintained for each individual registered with the system. The presence server supports presence clients that register with the server, including TeamPortal, ConnectIcon, and OpenChannel clients, as described earlier. Presence clients are Java classes that run remotely and can both report changes in presence status and react to presence server messages, telling them that the presence information has changed for a user they have defined as being in their awareness set. Servlet solutions are being investigated to enable presence services in a wireless environment.

In addition, a small application detects user presence activity on the soft endpoint. On a PC, these activities are keystrokes and mouse events. Messages are sent from the application to the presence server when the system state changes from being idle to detecting use of the keyboard/mouse or when detecting no activity.

Service Creation and Third-Party Programmability

A service creation environment compatible with the Lucent Full Circle API is being created for SAMM.[6] Third-party programmability will be supported through extensible APIs at both the messaging and programming levels. The extensibility characteristics of SAMM will allow third parties to evolve the frameworks and to add new services on a live system. Service executables may reside on SAMM or on third-party application servers. Third-party applications may use network resources or undertake interactions with external entities. Programmable policy agents will constrain the usage of the network by third-party services. The service component classes and OAM&P abstract classes may be customized by third-party subclasses. The service components may interface to:

- An intelligent network (IN) service control point (SCP) via transaction capability application part (TCAP)[11]/Signaling System 7 (SS7) or TCAP/IP;
- The new generation of application servers via Java APIs for Integrated Networks (JAIN*),[12] Parlay,[13] Extensible Markup Language (XML),[14] or SIP;
- Lucent's PacketIN™[15] computer software via JTAPI, JAIN, or Parlay.

Third parties may develop new services by creating or adapting components on SAMM or by programming across open APIs.

Programming new components on SAMM. A third party may choose to develop a new service at the service component level by writing subclasses of service components and OAM&P classes. These subclasses may interface with new classes unrelated to those provided by Lucent, empowering them with limitless possibilities for creating new services. New core service logic classes may be plugged into the service logic component bus during service deployment. The new core service logic will be applicable across the access types and transport protocols. The service provider may also write new device server components when introducing new access types or new transport protocols, expanding the capabilities of SAMM to follow new standards and new technology curves.

Programming across open APIs. Third parties may also choose to develop value-added services at a higher level—that is, across the open APIs, located on SAMM or application servers. The open APIs include JTAPI and XML-based APIs. In the future, JAIN, Parlay, TCAP/SS7, and TCAP/IP will also be supported. The service framework classes may invoke these services across a distributed processing environment through the communications middleware. Sample services could include the new TelePortal service[16] using either an XML-based interface or Web-based call center services. The TelePortal service allows Web servers to host telephone access to speech-enabled voice services, providing capabilities such as text-to-speech, automatic speech recognition, and audio file play and record. For example, Web-based call center services could include application servers that host automatic call distribution and assisted browsing.

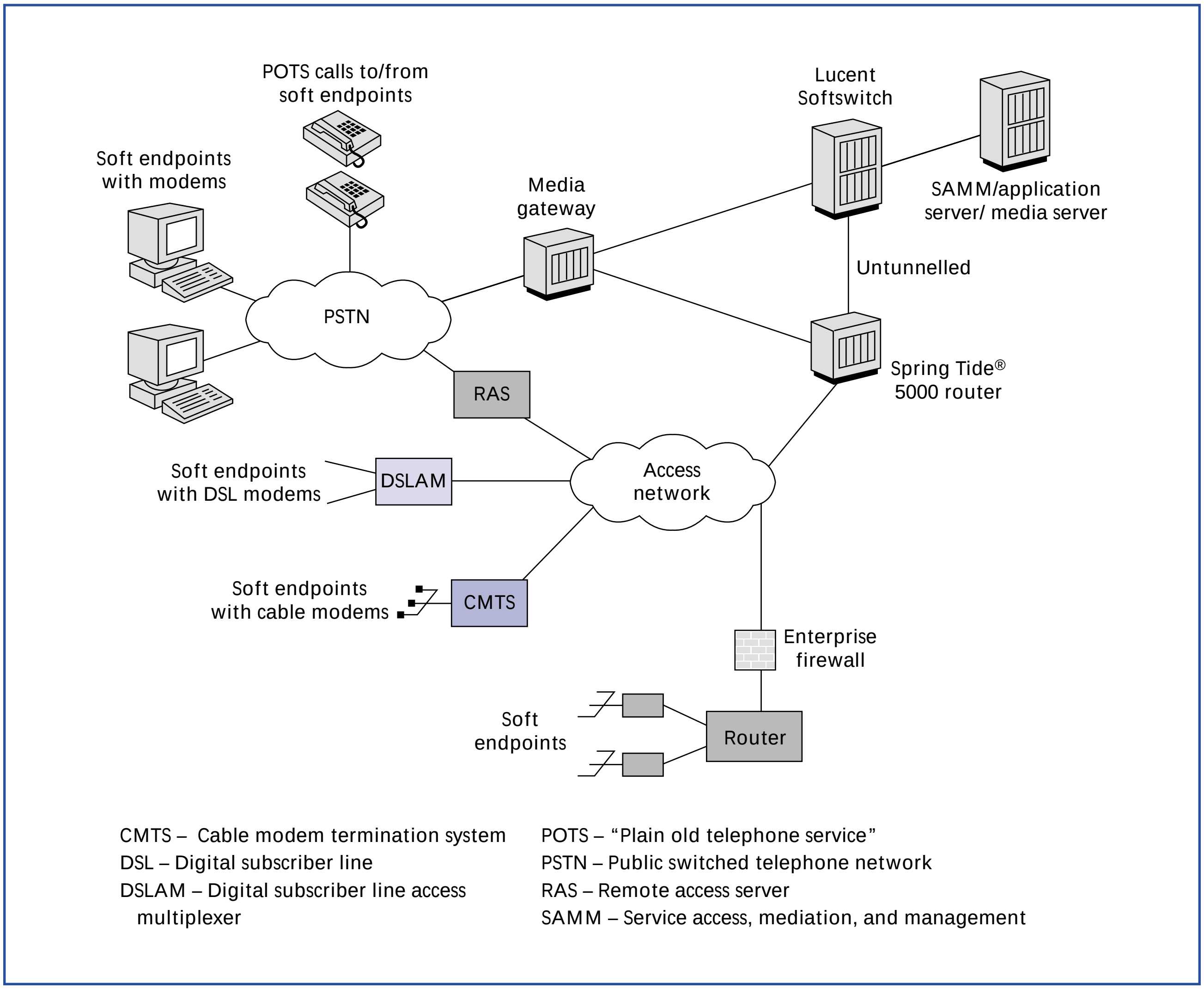


Figure 5.
Current SAMM configuration.

SAMM in the Network

Along with SAMM, the Lucent Softswitch provides services to softphones with SIP and H.323 clients, which perform service signaling for SAMM through a packet access network. Calls among the softphone users route through the packet network, while calls between SIP and H.323 clients are interworked through Lucent Softswitch SIP and H.323 device servers. Calls set up between a softphone and a phone terminated in a PSTN end office use SS7 signaling through the Lucent Softswitch SS7 device server, with bearer traffic flowing through a media gateway to SS7 trunks and then to the PSTN.

The initial SAMM network architecture supports connectivity to PSTN end offices for H.323 soft-endpoint-to-PSTN calls. **Figure 5** shows the SAMM network architecture in the current configuration.

Calls interwork with the PSTN through a media gateway, which converts between IP and PSTN formats. The media gateway supports SS7 trunks to the PSTN end offices, which can be the Lucent 7R/E™ Now switch or other vendor switches. The SS7 trunks may be E1 trunks for international applications or T1 trunks for North American region (NAR) deployment. The media gateway will need to support VoIP calls originating from remote places and entering via modems, as shown in Figure 5.

Calls interwork with the access network through an IP router, such as the Spring Tide® 5000 router.[17] An IP router provides network-based firewall services, user identification, and authorization; it also contains a platform for usage-based billing, dynamic virtual private network (VPN), tiered service selection, and customer network management.

Service Scenarios

The service scenarios and call flow diagrams described in this section demonstrate how the end user employs the collaboration services; they also illustrate how the service framework architecture objects work together to provide collaboration services.

The software objects are grouped into the four major functional categories depicted in the component architecture—the services framework, application server, media server, and Lucent Softswitch. The fifth category represents the soft endpoint. In **Figures 6** and **7**, the scenario call flow diagrams, the complete set of software objects are not explicitly shown, and a few of the objects are abstracted into functional groupings.

Each scenario includes some underlying assumptions:

- All users are registered users of the collaboration services, and the presence server maintains information concerning each user's preferred device for receiving voice calls.
- Each user's preferred device for receiving voice calls could be the existing circuit phone on his or her desk, or a VoIP soft endpoint, or a mobile phone.
- The user can receive voice calls on any device, regardless of the format in which the call was initiated.
- All users have a soft endpoint environment, meaning each user has a browser that supports Java applets.
- The Lucent SoftSwitch is provisioned to trigger JTAPI for call processing of all phone numbers associated with users.
- A small presence client application is installed on each user's PC to send presence information to the presence server.

Scenario Overview

The service scenarios provide a context for how the collaboration services might be used by a customer in a service provider network. These service scenarios support the efforts of the following workers in a company:

- Al, a field representative;
- George, an R&D manager;
- Pat, the marketing manager,
- Kathy, a member of the Strategy Group;
- Ann, a business consultant who works from home; and
- Lalita, a director.

All workers are registered users of the converged collaboration services.

Scenario Description

In this scenario, shown in Figure 6, Al has just finished talking to a customer. Al has a great idea for a new product offering, which he would like to share with George, the R&D manager, and Lalita, the director. Al sends a ConnectIcon message to George and Lalita, with a link to a proposal. As shown in the call flow of Figure 6, the ConnectIcon information is sent to the presence server and is then stored in the data associated with each user.

Lalita views her TeamPortal page and immediately receives the following ConnectIcon information:

- She sees the ConnectIcon in her TeamPortal area.
- She sees that Al is available, but George is not.
- She accesses the document.
- The document is then loaded into her browser via a standard hypertext transport protocol (HTTP) query.

Lalita reviews the document and clicks an "I'm Prepared" button in her ConnectIcon, which sends a "Prepared" message to the presence server. This "Prepared" status is shared with the team via the TeamPortal.

George is unavailable. When George returns and logs on, his TeamPortal notifies the presence server. George sees his ConnectIcon message, accesses the document, reviews it, and clicks the "I'm Prepared" button in his ConnectIcon.

The ConnectIcon on Lalita's TeamPortal receives notification that George has become available and dis-

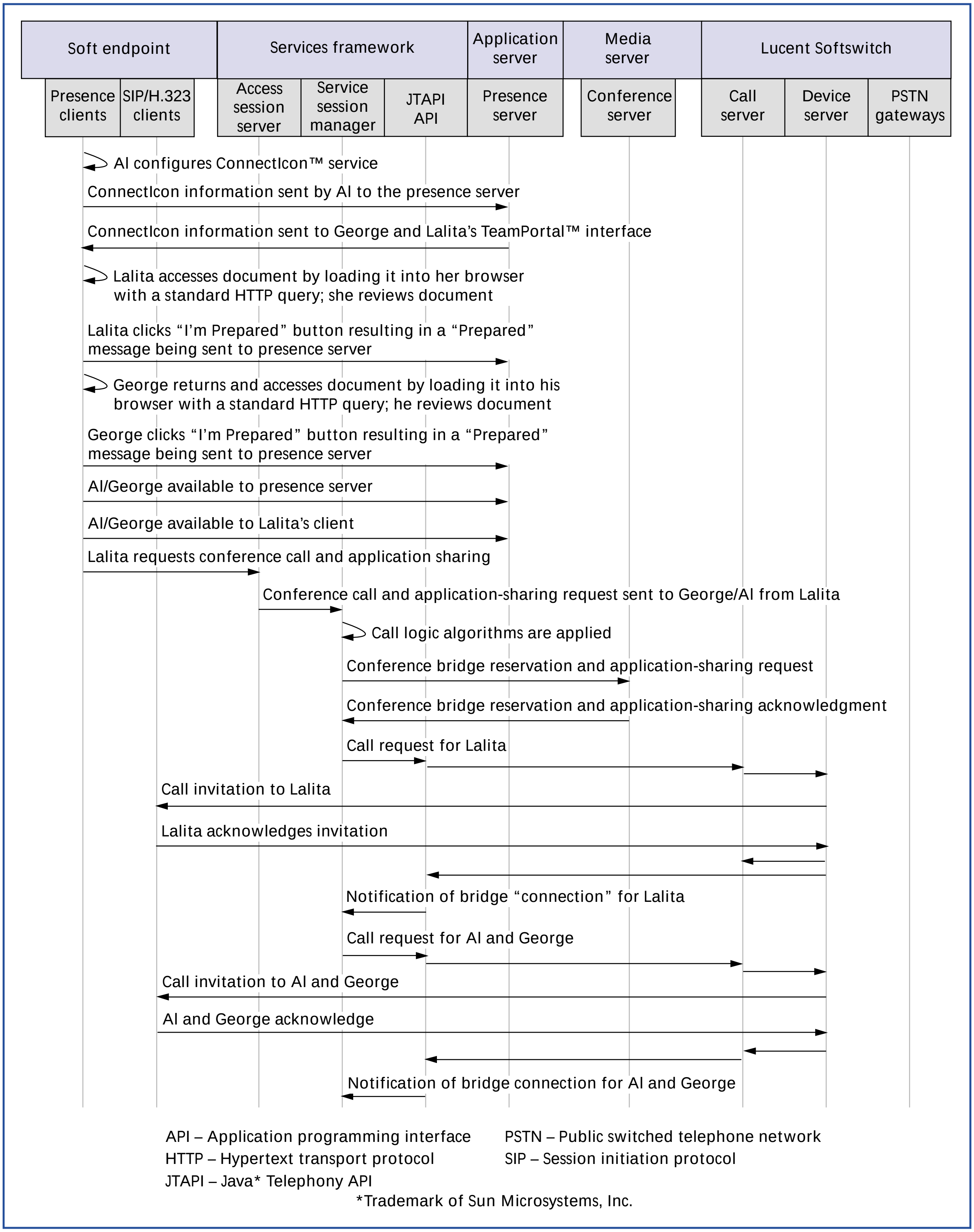


Figure 6.
ConnectIcon™ scenario call flow.

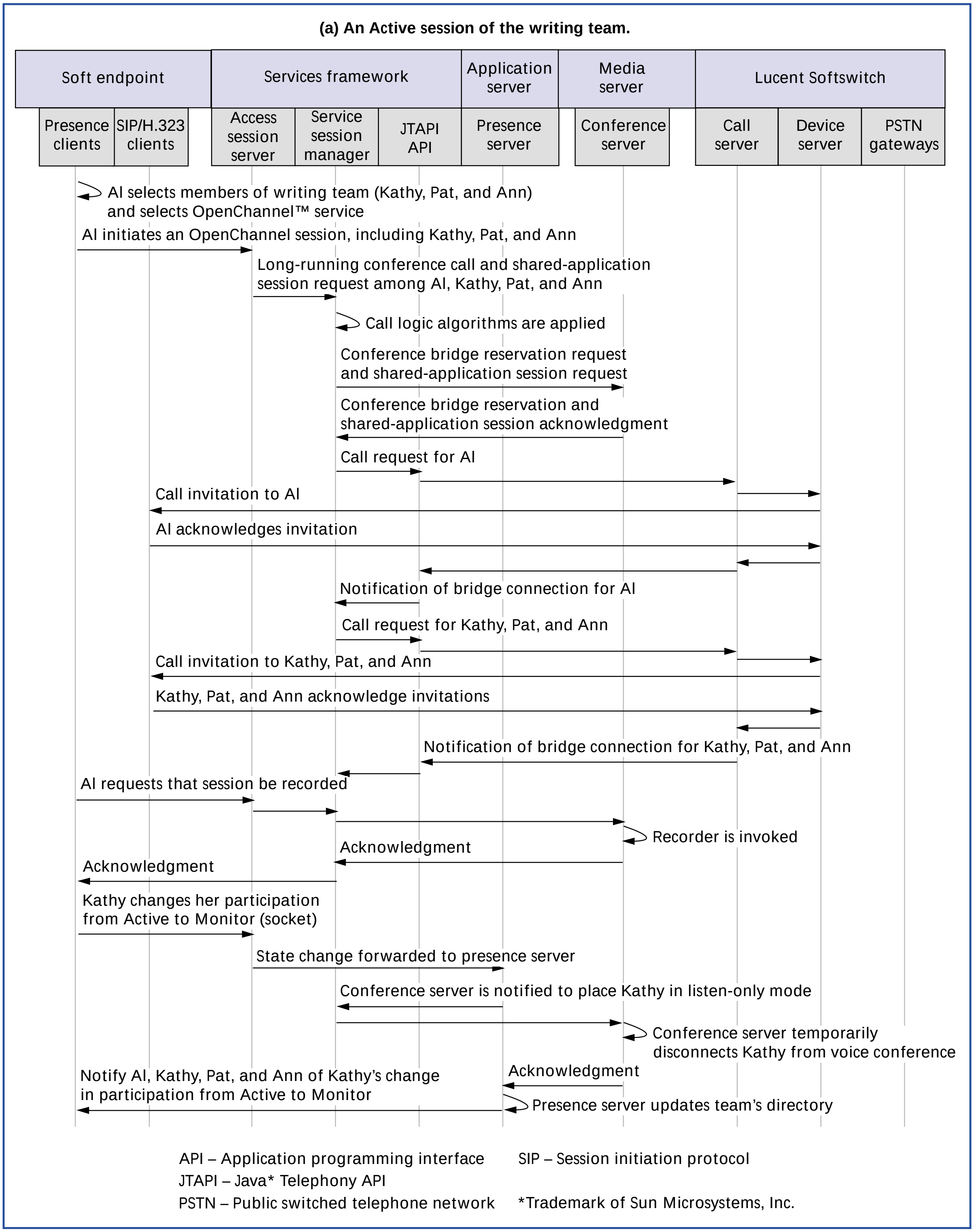


Figure 7.
OpenChannel™ scenario call flow.

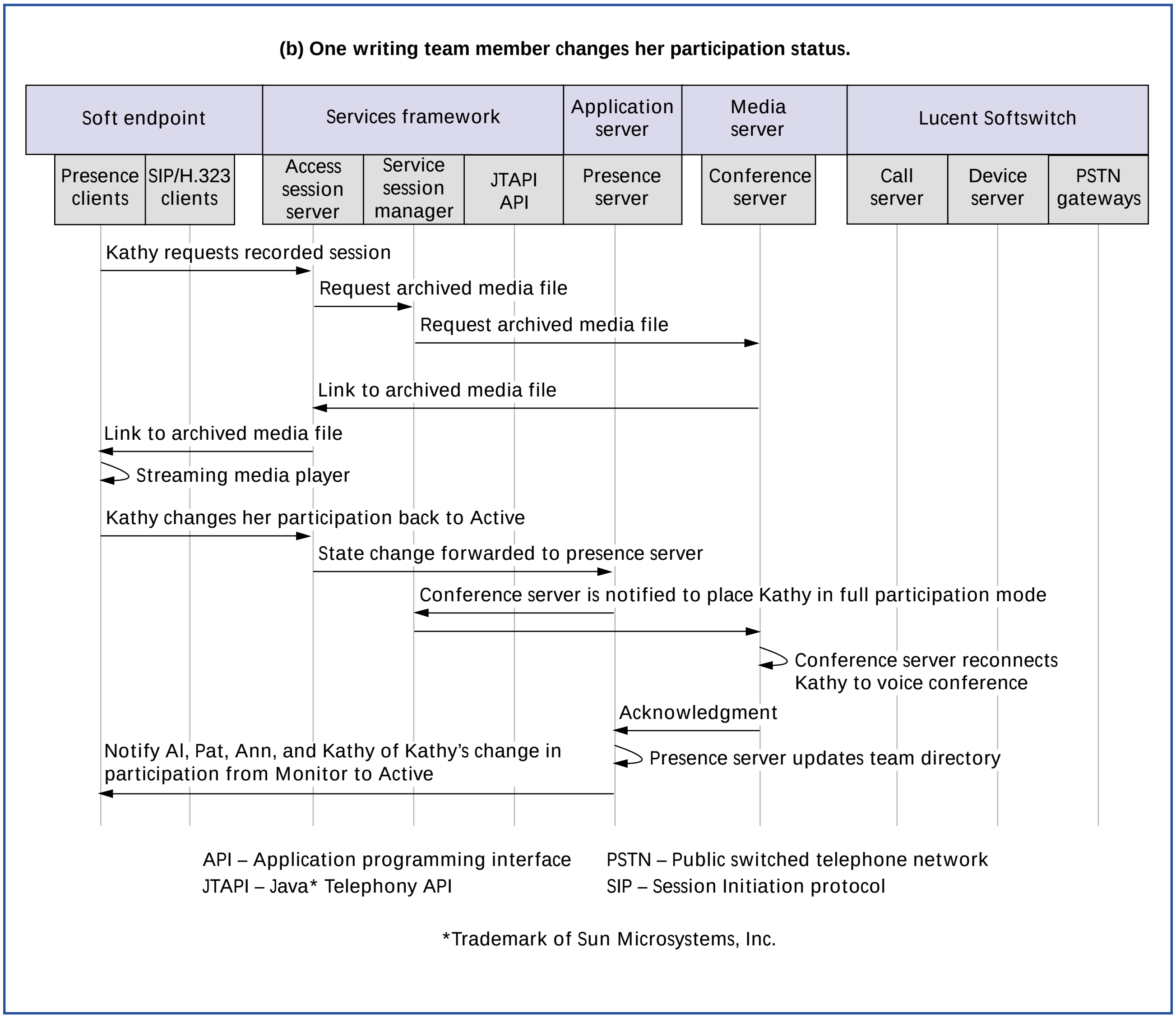


Figure 7 (continued).
OpenChannel™ scenario call flow.

plays this information. It also indicates that Al is still available. Using click-to-dial, Lalita initiates a call and an application-sharing session with George and Al to review the plan.

Figure 6 depicts this exchange, showing the communication flow among the architecture's components.

George and Lalita love Al's idea and want a complete business plan in five days. Al forms a writing team, which includes Kathy, Pat, and Ann. The team decides to use OpenChannel service and to record the session. The text below and Figure 7 describe the OpenChannel session and the information flow among the architecture components that support this scenario.

Using his TeamPortal, Al initiates an OpenChannel session by selecting the members of the writing team in TeamPortal and then selecting "OpenChannel," as shown in Figure 7a.

- Al's TeamPortal requests that the access server establish a (long-running) audio connection involving all members of the team. The access server forwards the request to the service session manager. After the call logic algorithms are applied in the appropriate service session modules, the service session manager sends the reservation request to the conference server.

- The conference server acknowledges the reservation request and provides the information required to complete the calls concerning the conference bridge.
- The service session manager initiates Al's voice call setup, taking into account his preferred device, which in this example is his softphone.
- Al's softphone is connected to the conference bridge.

Once Al's PC has acknowledged connection to the conference bridge, call requests and connections are also made to Kathy, Pat, and Ann. The OpenChannel session is now active and the team decides to start recording the session. Al initiates the record request.

For the morning of Day 2, Kathy changes her participation from Active to Monitor in order to join another strategy session.

- Kathy's TeamPortal notifies the presence server about the change in her participation state. The presence server then updates the team directory and notifies the other members of the business plan team. Kathy's participation is now identified as Monitor on the OpenChannel frame for all team members.
- The OpenChannel is modified when the conference server is notified to place Kathy in listen-only mode.

Over her lunch, Kathy reviews the recording of the morning session and then changes her participation state to Active, as shown in Figure 7b.

- Kathy's TeamPortal requests that the access server provide a link to the appropriate archived media file. The access server forwards the request to the service session manager, who sends the request to the conference server.
- The conference server provides the link to the access session, which forwards it to the TeamPortal. TeamPortal makes it available to the streaming media player on Kathy's softphone.
- Kathy directs the streaming media player to play, fast forward, and pause for her review of the morning session.
- Kathy's TeamPortal sends a notification of the change in her participation state to the access session server, which initiates the state change process. Kathy's participation is now identified as Active on the OpenChannel frame for all team members.
- The actual OpenChannel is modified as the conference server is notified to place Kathy in full participant mode.

The service scenarios and call flow diagrams in this section have described how end users might use collaboration services and illustrate how the service framework architecture objects work together to provide collaboration services. We presented an example of distributed work under short time schedules; the work is supported by presence information, long-running conference calls, and shared applications.

Conclusion

As voice and data networks converge, Lucent must provide new revenue-generating services that integrate modes of communication, going beyond unified messaging and click-to-dial service. We have described new services that combine and extend presence awareness and instant messaging, exploiting all the information about users in the network. SAMM takes advantage of all available means to establish communication, including POTS, text-based messaging, and VoIP using softphones and SIP. These services allow variable levels of participation, while engaging and integrating all necessary communication channels. They also allow easy transition from asynchronous (such as e-mail) to synchronous (such as conference call) communication.

Our goal is to change the concept of staying in touch across distances, to go beyond the circuit-driven dichotomy of on-hook or off-hook. We described TeamPortal, ConnectIcon, and OpenChannel, a suite of Bell Labs collaboration services built on SAMM. SAMM exposes presence data in the network to registered users, integrates different modes of communication, and supports extended service logic, thus enabling new, innovative converged services.

Acknowledgments

We are grateful to David Atkins, Chris Chrin, S. Carolyn Darity, Mark Handel, Janet Dianda, Bin Ho, Lalita Jagadeesan, Kristin Kocan, Steven Kovarik, Tsuey-ru Lu, William Opdyke, Prashant Parikh, and Kim Scott, each of whom contributed to the success of this project.

References

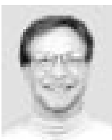
1. J. R. Dianda, B.-W. Ho, and K. F. Kocan, “Reducing Complexity for Converged Voice/Data Networks and Services Architecture,” *Bell Labs Tech. J.*, Vol 5, No. 2, Apr.–June 2000, pp. 55–71.
2. J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, “Distance, Dependencies, and Delay in a Global Collaboration,” *Proc. of CSCW 2000*, Philadelphia, Pa., Dec. 2–7, 2000, pp. 319–328.
3. Lucent Technologies, “Lucent Technologies Softswitch Next Generation Network Solutions… Now,” <http://www.lucent.ssg.com/ons/softswitch/>.
4. Lucent Technologies, “Full Circle™ Program,” <http://www.lucent.com/full_circle/home.html/>.
5. B. Nardi, S. Whittaker, and E. Bradner, “Interaction and Outeraction: Instant Messaging in Action,” *Proc. of CSCW 2000*, Philadelphia, Pa., Dec. 2–7, 2000, pp. 79–88.
6. J. R. Dianda, S. C. Darity, B.-W. Ho, and K. J. Scott, “Service Authoring for Third-Party-Programmable, Service-Mediation-Enabled Feature Servers in the Multiservice Core,” *Bell Labs Tech. J.*, Vol. 6, No. 1, Jan.–June 2001, pp. 192–210.
7. M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, “SIP: Session Initiation Protocol,” Internet Engineering Task Force, IETF RFC 2543, Mar. 1999, <http://www.ietf.org/rfc/rfc2543.txt/>.
8. International Telecommunication Union, “Packet-Based Multimedia Communications Systems,” ITU-T Rec. H.323, Feb. 1998, <http://www.itu.int/itudoc/itu-t/rec/h/h323.html/>.
9. Sun MicroSystems, Inc., “Java™ Telephony API, Version 1.3,” <http://java.sun.com/products/jtapi/index.html/>.
10. W. Yeong, T. Howes, and S. Kille, “Lightweight Directory Access Protocol (LDAP),” Internet Engineering Task Force, IETF RFC 1777, Mar. 1995, <http://www.ietf.org/rfc/rfc1777.txt/>.
11. International Telecommunication Union, “Transactions Capability Application Part (TCAP),” ITU-T Rec. Q.771–Q.775, June 1997, <http://www.itu.int/itudoc/itu-t/rec/q/q500–999/index.html/>.
12. Sun Microsystems, Inc., “JAIN™ APIs for Integrated Networks,” <http://java.sun.com/products/jain/index.html/>.
13. The Parlay Group, Specifications, <http://www.parlay.org/specs/index.asp/>.
14. World Wide Web Consortium, “Extensible Markup Language (XML),” Version 1.0, W3C Rec. 10, Feb. 1998, <http://www.w3.org/TR/1998/REC-xml-19980210/>.
15. Lucent Technologies, “Introduction to PacketIN™ Solutions,” <http://www.lucent.com/IN/pin_intro.html/>.
16 Lucent Technologies, “TelePortal™ Solution for Speech-based Web Services,” <http://www.lucent.com/IN/pin_tp.html/>.
17. Spring Tide Networks, Inc., “Powering the Business-Quality Public IP Network,” <http://www.springtidenet.com/products.html/>.

(Manuscript approved April 2001)

RAYMOND O. COLBERT is a technical manager in the Resource Server Solutions Department of Lucent’s Switching Solutions Group in Naperville, Illinois. Currently, he is working on the 7R/E™ multimedia resource server and converged voice/data services and solutions. Mr. Colbert holds an A.B. degree in mathematics and physics from Illinois College in Jacksonville and B.S. and M.S. degrees in electrical engineering from the University of Illinois at Urbana-Champaign. He has five patents in the areas of hardware design, physical design, and services.

DIANE S. COMPTON, a product manager in IP Services Offer Management, part of Lucent’s Switching Solutions Group in Naperville, Illinois, is responsible for finding and developing market opportunities in the area of next-generation services. She received a B.A. in computer science and English from North Central College in Naperville, Illinois, and an M.S. in executive communications from Northwestern University in Evanston, Illinois.

RANDY L. HACKBARTH is a technical manager in the Software Production Research Department of Bell Labs, part of Lucent Technologies in Naperville, Illinois. His primary work responsibilities are to help establish, coordinate, and contribute to business unit/research partnership projects, especially in the area of next-generation communication systems. Mr. Hackbarth earned a B.S. in mathematics from Valparaiso University in Indiana, as well as an M.A. in mathematics and an M.S. in computer science from the University of Wisconsin at Madison.



JAMES D. HERBSLEB is a member of technical staff and a leader of the Bell Labs Collaboratory project in the Software Production Research Department of Bell Labs in Naperville, Illinois. He holds a B.A. degree in economics and psychology from Monmouth College in Monmouth, Illinois, an M.S. degree in computer science from the University of Michigan in Ann Arbor, and a Ph.D. degree in psychology from the University of Nebraska in Lincoln. For the past 10 years, Dr. Herbsleb has conducted research in the areas of collaborative software engineering, human–computer interaction, and computer-supported cooperative work. More recently, his work has focused on collaboration technology that supports large globally distributed projects.

LAURIE A. HOADLEY is a member of technical staff in the Switching Architecture, Performance, and System Engineering Department of Lucent's Switching Solutions Group in Naperville, Illinois. She is a member of the Switching Architecture Evolution Planning Team. Her responsibilities include trial and demonstration architecture support, as well as identification and integration of converged services and applications on the SAMM platform. Ms. Hoadley received B.S. and M.S. degrees in statistics from the University of Illinois at Urbana-Champaign.

GRAHAM J. WILLS, a member of technical staff and lead researcher in the Software Production Research Department of Bell Labs in Naperville, Illinois, received a B.A. degree in mathematics and a Ph.D. in statistics from Trinity College in Dublin, Ireland. He is currently working on Web-based visual environments that can be customized to specific domains. Dr. Wills pioneered the development of a C++ visualization library that formed the basis of the Visual Insights venture and also worked on information visualization for general data analysis, fraud detection, very large network analysis, knowledge discovery and clustering, real-time systems, and software and logfile analysis. ◆