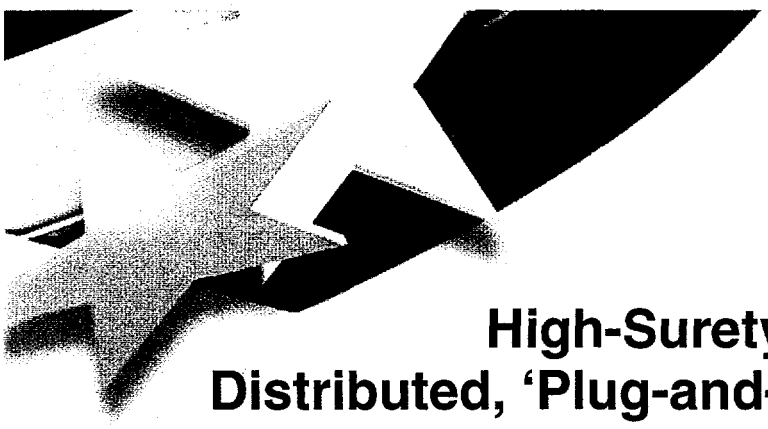
March 17, 1999



High-Surety Telemedicine in a Distributed, 'Plug-and-Play' Environment

Richard L. Craft, M.S.,[1] Steve Warren, Ph.D.,[2] Raymond C. Parks, C.C.P.,[3] Linda K. Gallagher, B.S.,[4] Rudy J. Garcia, M.S.,[5] and Donald R. Funkhouser, M.S.[6]

[1]Information Systems Surety Department, [2]Simulation Technology Research Department, [3]Distributed Systems Assurance Department, [4]Software Surety Department, [5]International Infrastructure Surety Department, [6]Decision Support Systems Architectures Department, **Sandia National Laboratories, Albuquerque, NM**

Keywords: telemedicine, security, plug-and-play, interoperability, architectures, components

Abstract

Commercial telemedicine systems are increasingly functional, incorporating video-conferencing capabilities, diagnostic peripherals, medication reminders, and patient education services. However, these systems (1) rarely utilize information architectures which allow them to be easily integrated with existing health information networks and (2) do not always protect patient confidentiality with adequate security mechanisms. Using object-oriented methods and software wrappers, we illustrate the transformation of an existing stand-alone telemedicine system into 'plug-and-play' components that function in a distributed medical information environment. We show, through the use of open standards and published component interfaces, that commercial telemedicine offerings which were once incompatible with electronic patient record systems can now share relevant data with clinical information repositories while at the same time hiding the proprietary implementations of the respective systems. Additionally, we illustrate how leading-edge technology can secure this distributed telemedicine environment, maintaining patient confidentiality and the integrity of the associated electronic medical data. Information surety technology also encourages the development of telemedicine systems that have both read and write access to electronic medical records containing patient-identifiable information. The win-win approach to telemedicine information system development preserves investments in legacy software and hardware while promoting security and interoperability in a distributed environment.

Contact:
Steve Warren, Ph.D., Principal Member of the Technical Staff, Sandia National Laboratories, P.O. Box 5800, M/S 1179, Simulation Technology Research Department, Albuquerque, NM 87185, Phone: (505) 844-4473, Fax: (505) 844-0092, Email: swarre@sandia.gov , Internet: http://www.sandia.gov

Prepared for ***Toward an Electronic Patient Record '99 (TEPR '99)***, May 1-6, 1999, Orange County Convention Center, Orlando, FL. Sponsored by the Medical Records Institute, 567 Walnut Street, P.O. Box 600770, Newton, MA 02460. Phone: (617) 964-3923, Fax: (617) 964-3926, Internet: http://www.medrecinst.com

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Table of Contents

Disclaimer of Liability

Introduction

The United States health care industry is experiencing a dramatic paradigm shift due to the convergence of several technology areas. Increasingly-capable telemedicine systems and the internet are not only moving the point of care closer to the patient, but the patient can now assume a more active role in his or her own care. These technologies, coupled with (1) the migration of the health care industry to electronic patient records and (2) the emergence of a growing number of enabling health care technologies (e.g., novel biosensors, intelligent software agents, and wearable devices), demonstrate unprecedented potential for effectively delivering highly automated, patient-centric health care while at the same time reducing the cost of care [1].

The increasingly functional commercial telemedicine systems available today incorporate video-conferencing capabilities, diagnostic peripherals, medication reminders, and patient education services. However, most of these commercial systems are custom-designed, "stovepipe" systems that do not inter-operate with other commercial offerings. Users are limited to a set of functionality that a single vendor provides and must often pay high prices to obtain this functionality, since vendors in this marketplace must deliver entire systems to compete. If the user desires additional features, they either pay a premium price for the necessary research and development, or they purchase an additional system that supports those features. Regardless, extra but unwanted functionality must be purchased as part of the "package deal." Most importantly, these systems (1) rarely utilize information architectures which allow them to be easily integrated with existing health information networks (HIN's) [2] and (2) do not always protect patient confidentiality with adequate security mechanisms.

Irrespective of the limitations of current telemedicine systems, an increasing number of environments are being defined that will require remote access to medical information, including patient-confidential electronic patient records (see Figure 1). In order to support these new environments, future telemedicine systems will need additional features:

- the ability to communicate with distributed objects,
- plug-and-play capabilities that allow telemedicine systems and peripherals to interact with those provided by other manufacturers,
- standard interfaces, communication protocols, messaging formats, and data definitions for interacting with electronic patient record databases, and
- most importantly, information surety technology to maintain patient confidentiality as well as the integrity, availability, and reliability of electronic medical information.

Sandia National Laboratories proposed a reference architecture for telemedicine [3] in order to promote plug-and-play interoperability between telemedicine systems and devices that populate a distributed, highly-networked environment. This paper discusses

- how to transform a stand-alone, stovepipe system into a plug-and-play system based on the Sandia architecture, and
- surety mechanisms that will exist within the architecture to protect patient confidentiality and the integrity of electronic medical data.

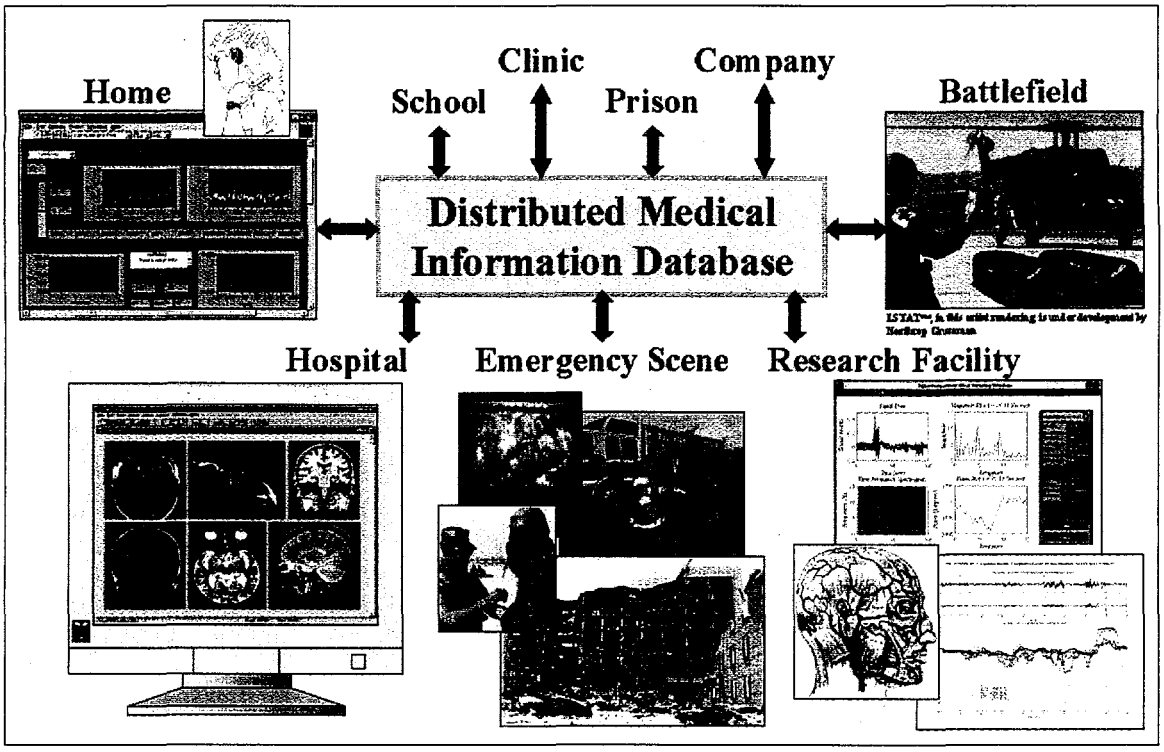


Figure 1. Environments that will require secure, reliable read/write access to medical information.

Overview of the Sandia Architecture

Areas of Functionality

In general, telemedicine devices provide seven types of services, depicted in Figure 2. Note that every device does not provide all of these services.

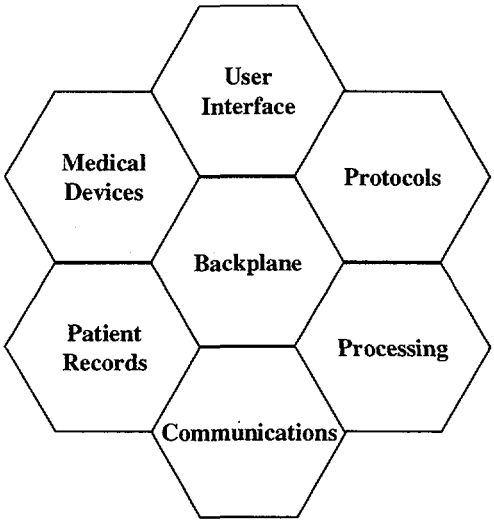


Figure 2. General types of services represented in the proposed telemedicine device architecture.

The following items describe these architectural services:

- The ***USER INTERFACE*** service represents hardware and software with which the user interacts, including mechanisms that support telemedicine device control (e.g., buttons and lights on an instrument display panel) and person-to-person interactions.
- The ***MEDICAL DEVICES*** service represents mechanisms for acquiring patient data, delivering therapy to a patient, or analyzing specimens collected from a patient.
- The ***PATIENT RECORDS*** service represents a device's ability to store and retrieve patient information that the device has collected.
- The ***PROCESSING*** service consists of specialized routines to manipulate data. Examples of this include statistical routines to analyze trends in data sets, filtering routines to manipulate waveforms and images, and "intelligent agents" that aid in diagnosis and care planning.
- The ***COMMUNICATIONS*** service represents (1) mechanisms a telemedicine device uses to interact with other devices and (2) services that support these communications (e.g., directories that indicate locations of specific services).
- The ***PROTOCOLS*** service constitutes the brain of a telemedicine device. The "programs" or "scripts" in this service area accomplish specific medical objectives by utilizing resources acquired from the other services. A simple protocol might, for example, direct a medical instrument to take a reading, tell the patient record to store the reading, and tell the user interface to display the reading. Protocols can deliver sophisticated functionality through command nesting.
- Finally, the ***BACKPLANE*** service represents mechanisms that tie the other six services together. It provides intra-device communications as well as profile information needed for device "self-awareness."

Converting a Stand-Alone System to a Plug-and-Play System that Utilizes the Sandia Architecture

Most commercially available, turnkey telemedicine systems are composed of tightly integrated collections of components. Adding new components and getting them to work seamlessly with the old components can prove to be difficult if not impossible. By contrast, future telemedicine systems will be based on plug-and-play technologies that use high-bandwidth communications to create systems composed of physically dispersed but cooperating components. Unfortunately, until then developers of telemedicine systems and owners of existing systems will face the problems associated with making "closed" or "non-standard" designs work in a world that is moving toward "open," standards-based designs.

This section describes work that Sandia National Laboratories is doing to help the telemedicine community identify plug-and-play approaches worth pursuing. Specifically, this section discusses the transformation of a stand-alone telemedicine system into a secure, plug-and-play system designed around the proposed Sandia telemedicine device architecture.

Block Diagram of a Stand-Alone Telemedicine System

In support of a telemedicine cost-effectiveness and diagnostic feasibility study performed by the Alton Ochsner Medical Foundation in New Orleans, LA, Sandia National Laboratories surveyed the telemedicine market and purchased a turnkey telemedicine system that delivered most of the video, sensor, and record storage capabilities needed for the study. The general system design is similar to that of other desktop telemedicine systems available today, although elements of its implementation separate it from other commercial offerings. A conceptual block diagram of the telemedicine system is depicted in Figure 3. While the various control routines are shown as separate elements within the "Application and Operating System" blocks, these elements are actually rendered as single, monolithic applications. Similarly, the instruments in the patient unit are tightly integrated with each other (although they can be removed as a unit from the rest of the system).

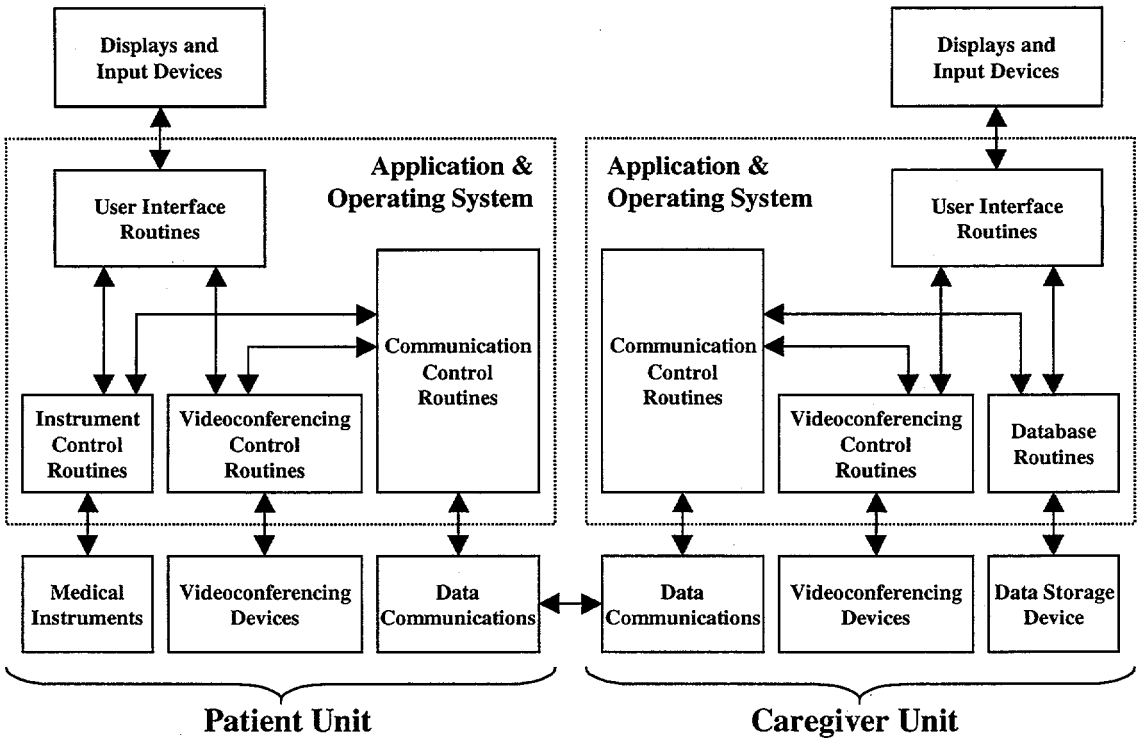


Figure 3. Conceptual block diagram of a stand-alone commercial telemedicine system.

Component Interactions in the Reference Telemedicine Device Architecture

In reworking the conceptual design of the target telemedicine system, Sandia partitioned the system functionality into the services shown in Figure 2, adding mechanisms to enable full plug-and-play operation. The latter mechanisms dictate that components in each partition behave as shown Figure 4.

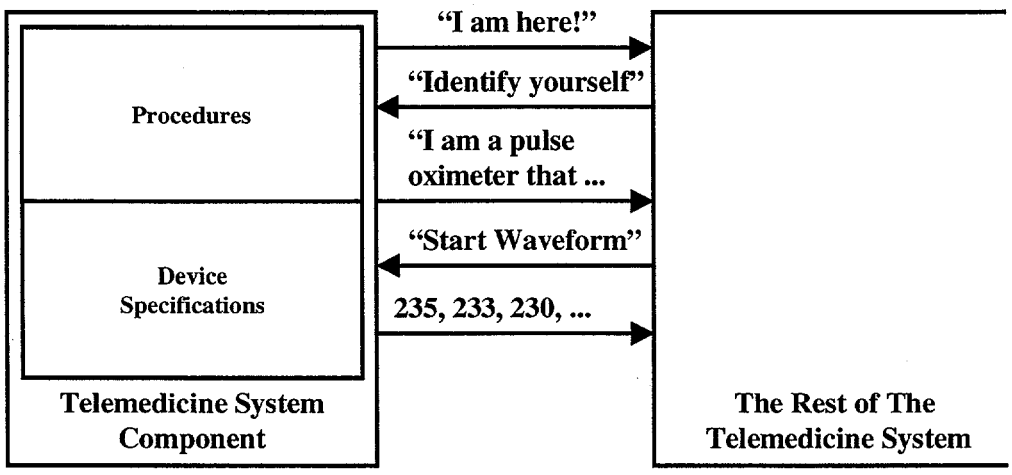


Figure 4. Component/system interactions that occur when a new device is added to the system.

Each component in the plug-and-play system is "self-aware," meaning that it can advertise both its type (e.g., a user interface device) and its capabilities (e.g., the ability to display both scalar values and time-

based light reflectance data from a pulse oximeter) using device specifications stored inside the component. When the component is added to a telemedicine system, the system recognizes its presence and requests that the component identify itself. In response, the component enumerates its capabilities. Information regarding these capabilities is stored in a backplane registry for later use. When the system executes an operation requiring those component capabilities, it sends messages to the component, which then completes the operation via procedures internal to itself (and perhaps unknown to the rest of the system). The component then uses a messaging mechanism to return data to the rest of the system.

Application-Level and Device-Level Interfaces

Each component defines two levels of interfaces. The first is an application-level interface, where messages passed between components focus on application domain concepts. For example, a sphygmomanometer would respond to commands like "initiate a blood pressure reading" and therefore produce relevant medical data (e.g., a systolic/diastolic/mean blood pressure triplet). At the application level, every device of a given type complies with a virtual device template defined for that class of device. To the degree possible, templates share common commands and data types (e.g., "start," "scalar result: ###").

Device-level interfaces support devices that are not inherently plug-and-play. At this level, the unique attributes of a physical device express themselves. For example, many commercial medical devices have serial interfaces with proprietary command protocols and data sequences. Using these protocols and sequences as a basis for integrating devices into telemedicine systems undercuts the more general goal of plug-and-play interoperability. This layer makes these legacy devices compliant with the plug-and-play model.

Figure 5 illustrates this dual-layer concept. All three devices in Figure 5 present the same application-level interface to the telemedicine systems they support. However, the following facts are hidden from these telemedicine systems: (1) device A employs vendor A's equipment and procedures, (2) device B employs vendor B's equipment and procedures, and (3) system C is a native-mode, plug-and-play device.

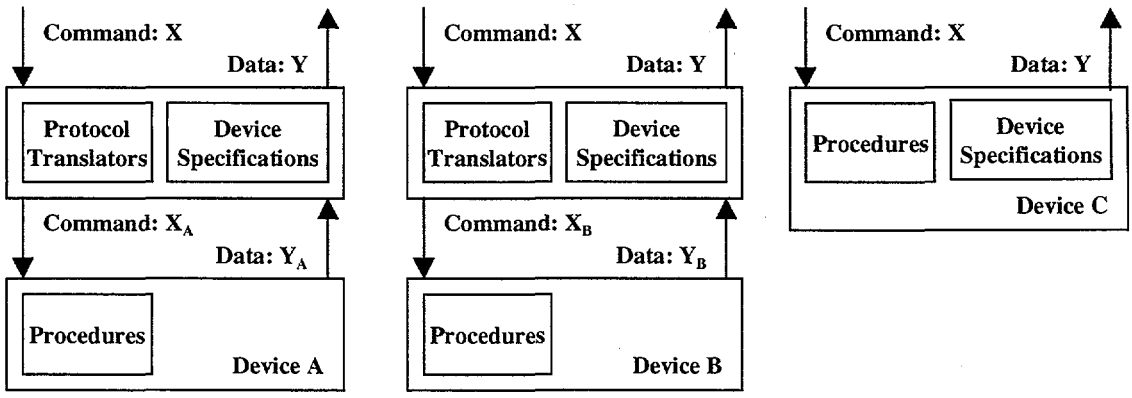


Figure 5. Application-level and device-level interfaces.

Relationship Between Interface Levels and the Architecture Service Areas

This two-level interface scheme expresses itself differently in each of the seven service partitions inside the telemedicine device architecture. At the application level, the telemedicine device architecture's **User Interface** is perceived by the other partitions as presenting sets of controls that provide medical functionality. For example, a sphygmomanometer might include two text display controls that present

blood pressure readings, a control to dictate instructions, and a collection of controls to start, stop, and calibrate the instrument. These controls would instruct the user and respond to user actions by sending messages to the rest of the telemedicine device that convey the intent of the user (e.g., "initiate blood pressure reading"). While the application-level interface to the set of User Interface controls would be standard, the mechanisms that implement those controls could vary greatly.

Medical Devices are either monolithic or partitioned into two layers, as shown in Figure 5. Either way, the application-level interface presented to the rest of the system denotes a standard medical device that accepts commands (e.g., "initiate blood pressure reading") and returns standard responses (e.g., "diastolic: XXX, systolic: YYY"). As depicted in Figure 5, the other parts of the system do not know whether this functionality is delivered by an integrated device or a set of components wrapped with a standard interface. In addition, the technologies used to deliver the requested services may be transparent (e.g., whether a blood pressure reading is obtained using a cuff or an intra-arterial catheter).

Patient Records present application-level interfaces for storing and retrieving patient-encounter data. This includes the capability to determine whose patient records are stored on a device, where records are stored for a given patient, and how these data can be accessed. As before, the delivery mechanism (e.g., an object database, relational database, or text file) can be transparent to the other components in the system. Similarly, the storage mechanism (e.g., disk drive or smart card) can be transparent to the components that read and write data to this media.

The abstract model used by the **Communications** partition separates the application-level services (primarily provision of bandwidth and directory facilities for locating support information) from the device-level mechanisms that actually move bits. Communications components provide single interfaces to the rest of the system for (1) requesting that connections be established with remote devices and (2) requesting that channels of a certain type (e.g., an internet protocol link) be allocated. How these connections are created, what these channels look like, and the transport medium (e.g., telephone or satellite) are hidden from the rest of the system.

While many **Processing** components will be implemented as simple software modules that advertise their capabilities (e.g., similar to JavaBeans™), some components will be dedicated-function, hardware devices that present standard component interfaces to the rest of the system (e.g., a device dedicated to high-throughput ultrasound image processing that accepts a digital data stream and outputs processed images into a patient record).

The primary function of **Protocol** components is to establish inter-component connections needed during a given medical procedure. For example, a protocol for taking a patient's temperature would connect a thermometer output to the user interface control that would display the temperature reading. Once these connections were established, the protocol would back out the way and watch for key events from the other components that indicate the need to alter component connections. While many protocols will be simple, doing nothing more than connecting and disconnecting components, some will be intelligent.

Finally, the **Backplane** presents two kinds of interfaces to the rest of the system. The first is a registry interface which notes when components join the system, change their internal state, or leave the system. Protocol components use registry information to locate resources needed for their operations [3]. The second interface includes objects that support local communications. While many of these same objects are found in Communications components, the Backplane objects differ because they are always present rather than dynamically created and destroyed as needed.

Block Diagram of the Transformed System

Using this virtual-device approach to telemedicine system design, the patient unit block diagram shown in Figure 3 transforms into the block diagram shown in Figure 6.

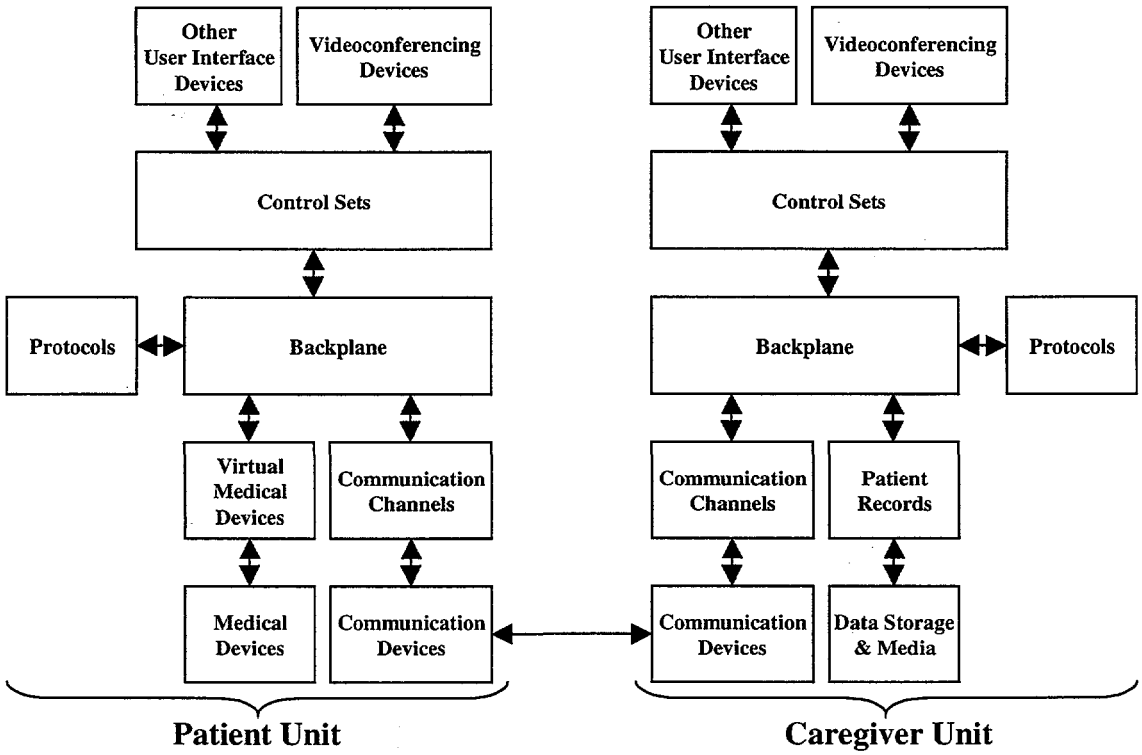


Figure 6. Block diagram of a telemedicine system after it was transformed to comply with the Sandia telemedicine device architecture.

In recasting a legacy telemedicine system to comply with the plug-and-play device architecture, Sandia wraps the software that controls the various system components (i.e., the graphical user interface, the medical instruments, the modem, and the videoconferencing equipment) with application-level interfaces so that they become virtual devices. The routines that control execution of specific medical functions (e.g., collecting and displaying a blood pressure reading) are embodied in protocols. The backplane registry and local communication functionality are added last.

The capabilities of a transformed system can be easily extended. For example, it is relatively straightforward to add modules that translate Health Level 7 [4] or COAS [5] queries into native commands. In addition, given the common-communication-channel construct used throughout the architecture, one can easily change the topology of the telemedicine system by adding and/or removing components from the system, thereby changing the way that functionality is partitioned among various devices.

Incorporating Information Surety

The previous sections described the telemedicine device architecture and the approach that one would use to modify an existing telemedicine system so that it would comply with that architecture. This section addresses information security in these plug-and-play systems. In a distributed system, security issues are significantly more complex than those encountered in traditional point-to-point systems. Distributed

telemedicine systems will not be wholly owned by one entity, but rather consist of dynamic alliances formed as-needed between components owned by multiple entities. This more complicated environment requires a radically different security model.

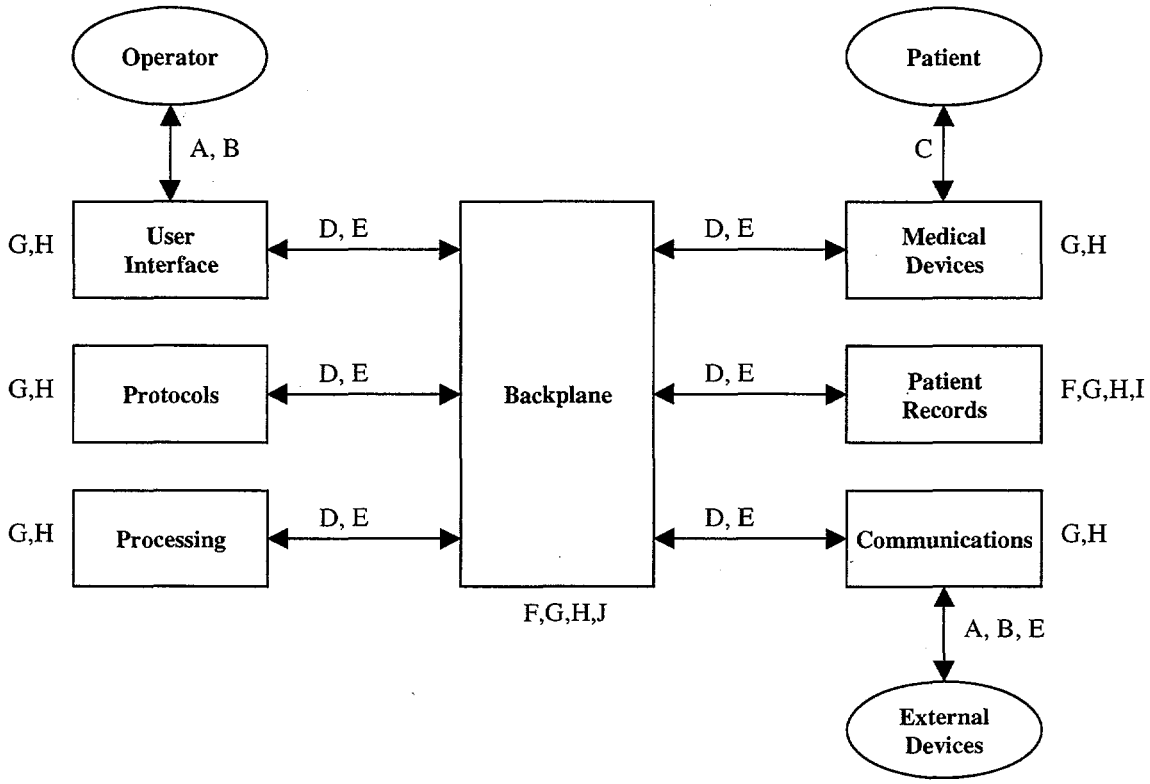


Figure 7. Points where security must be addressed in a distributed system.

In a distributed system made up of parties who trust (or do not trust) each other to varying degrees, one must address a number of security concerns. Figure 7 illustrates several areas where surety issues appear in the distributed telemedicine device architecture:

A. The telemedicine device must authenticate the identity and attributes (e.g., the role) of any person or entity that attempts to use the telemedicine device.
B. An operator or device must verify that they are sending commands to the correct device.
C. Since the telemedicine device is the agent of the caregiver with respect to operations on a patient, the association between the patient and the device must be authenticated.
D. In situations where component links and the supporting backplane are accessible to unauthorized entities, steps must be taken to ensure that the information flowing between these links is authentic, reflects the current intent of the sender (e.g., it cannot be replayed), and can be kept private.
E. Communication delivery should be assured. When communications are being thwarted, mechanisms should apprise the sender of this fact.
F. Patient records must be instantly available upon demand, not be disclosed to unauthorized users, or subject to corruption/deletion while in storage.
G. The telemedicine device must always operate correctly, depending on the criticality of the component in question, and must detect subversion or component failure.
H. Information that characterizes system components must not be subject to subversion.
I. The system must provide secure time stamping of patient record entries.
J. The telemedicine device should support key event auditing.

For stand-alone systems, each of these concerns can be problematic. However, for systems that rely on public networks, like the Internet, the increased connectivity amplifies these issues because of the much greater exposure to security threats.

Finally, in addition to these general requirements, a number of constraints must be levied on whatever solutions are considered. First, security solutions that protect telemedicine technology must be as open as the technology they protect. "Hard-wired" solutions that cannot evolve as the threat environment changes are unacceptable. Second, security solutions must meet the needs of individual sites. They cannot adhere to a "one size fits all" approach that forces sites with limited numbers of threats to employ the same mechanisms that protect sites exposed to a large number of threats. Third, solutions that are technology-specific should not drive the design of the security framework established for telemedicine devices. This is not to say that these solutions *cannot* be used. Otherwise, no security mechanism would be suitable. Instead, the intent is that solutions must permit, to the degree possible, transparent replacement of mechanisms that have outlived their usefulness.

To accommodate these needs, we are considering several key technologies. The first is encryption and cryptographic authentication, along with the requisite supporting services (e.g., key/certificate management, key generation). Because of its modular, open architecture, the Intel Common Data Security Architecture (CDSA) is the most likely candidate for this purpose [6]. For control of access to data and procedures, we are considering the object-oriented domain and type (OODTE) mechanisms currently researched by DARPA along with the Healthcare Resource Access Control (HRAC) facilities currently advanced in the CORBAmed community [7]. Suitable mechanisms for user authentication and auditing are still being investigated.

Conclusions

As health care information systems become distributed and mass-networked, telemedicine systems must adapt to function properly within that environment. In addition, for telemedicine systems to become cost-effective while providing better-quality health care, they must evolve into plug-and-play systems that utilize best-of-breed component technologies. However, in this distributed, componentized environment, information security becomes more problematic because multiple entities have control over the components that comprise these distributed systems.

This paper illustrated the transformation of a stand-alone telemedicine system into a componentized, plug-and-play system that supports the proposed Sandia telemedicine device architecture. Using this technique, commercial telemedicine systems and medical peripherals that were once incompatible can be upgraded to share data through standard interfaces and messaging protocols. Although security issues are more problematic in distributed telemedicine systems relative to point-to-point systems, those issues can be delineated and addressed individually. With health information surety mechanisms in place, telemedicine systems can maintain patient confidentiality and the integrity of electronic medical data, making them candidates for full read/write access with regard to electronic patient records contained in health information repositories. This win-win approach to telemedicine information system development preserves investments in legacy software and hardware while promoting security and interoperability in a distributed environment.

Acknowledgement

This work was supported by congressionally allocated funds administered by the Telemedicine and Advanced Technology Research Center, U.S. Army Medical Research and Materiel Command, Fort Detrick, Frederick, MD.

Author Biographical Information

Rick Craft is a Senior Member of the Technical Staff at Sandia National Laboratories, Albuquerque, NM, where he has worked for 14 years. For the majority of this time, Rick has analyzed and designed information systems, concentrating for the last five years on information security techniques as well as helping system designers identify and secure system vulnerabilities. Rick holds a B.S. and M.S. in Electrical Engineering from the Georgia Institute of Technology.

Steve Warren is a Principal Member of the Technical Staff at Sandia National Laboratories, Albuquerque, NM. He received a Ph.D. in Electrical Engineering from The University of Texas at Austin in 1994 and an M.S. and B.S. in Electrical Engineering from Kansas State University in 1991 and 1989, respectively. His technical interests include computational analysis in biomedicine, optical biosensor design, wearable sensors, advanced lighted-based diagnosis technologies, and object-oriented techniques applied to medical information systems. His current work focuses on (1) the design of plug-and-play systems for telemedicine and (2) particle transport simulations for determining photon dose distributions in human tissue.

Ray Parks is a Senior Member of the Technical Staff at Sandia National Laboratories, Albuquerque, NM. He received a B.S. in Engineering from the U.S. Air Force Academy in 1978. Ray's technical interests include distributed systems assurance, information warfare, object technology, middleware, intranets, and extranets. His current work focuses on red-teaming, vulnerability analysis, and the incorporation of security and surety into distributed systems.

Linda Gallagher is a Senior Member of the Technical Staff at Sandia National Laboratories, Albuquerque, NM. For the last 21 years, she has been developing software for nuclear safeguards and remote monitoring applications, typically focusing on user interface and database issues. Linda has a B.S. in Electronics Engineering Technology from the DeVry Institute of Technology in Chicago.

Rudy Garcia is a Senior Member of the Technical Staff at Sandia National Laboratories, Albuquerque, NM. Rudy primarily designs and implements data acquisition components for remote monitoring systems. For the last eight years, he has worked on a variety of embedded platforms and command-and-control systems. Rudy holds a B.S. and M.S. in Computer Engineering from the University of New Mexico and Boston University, respectively.

Don Funkhouser is a Principal Member of the Technical Staff at Sandia National Laboratories, where he has worked since 1982. He has a B.S. and M.S. in Electrical Engineering from Oklahoma State University. Don's previous experience focused on hardware and software design for embedded systems. More recently, he has concentrated on the development of object-oriented software systems and the use of software components in distributed applications.

References

1. **Strategies for the Future: The Role of Technology in Reducing Health Care Costs**, ©1996, Sandia National Laboratories, SAND 60-2469, DOE Distribution Category UC-900, November 1996. Available electronically at http://www.matmo.org/pages/library/papers/papers.html.
2. Jahsman, William E. "Integrating Telemedicine and the HIN: Adventures in Convergence – Let's Get Our Wires Uncrossed!," *Telemedicine Today*, February 19, 1999, pp. 26-30.
3. Warren, Steve, Richard L. Craft, Raymond C. Parks, Linda K. Gallagher, Rudy J. Garcia, and Donald R. Funkhouser. "A Proposed Information Architecture for TeleHealth System Interoperability," Paper to be presented at *Toward An Electronic Patient Record '99 (TEPR '99)*, Orange County Convention Center, Orlando, FL, May 1-6, 1999. Proceedings will be available through the Medical Records Institute, 567 Walnut Street, P.O. Box 600770, Newton, MA 02460. Phone: (617) 964-3923, Fax: (617) 964-3926, Internet: http://www.medrecinst.com.
4. Health Level 7, http://www.hl7.org/.
5. Clinical Observation Access Service, CORBAmed, http://developer.intel.com/ial/security/index.htm.
6. Common Data Security Architecture (CDSA), http://developer.intel.com/ial/security/index.htm.
7. CORBAmed, http://www.omg.org/CORBAmed