Connectivity

Building the Foundation for Medical Device Plug-and-Play Interoperability

Medical device communication standards are works in progress and hold the promise of universal communication among medical electronic devices and information systems.

Richard Schrenker and Todd Cooper

Think about the plethora of medical devices surrounding the patients in a hospital intensive care unit (ICU). A patient connected to one or more vital-signs monitors may also be receiving drugs or other fluids under the control of an infusion pump. More-acutely-ill patients may have some of their physiological processes supported by devices such as ventilators—in addition to the monitors and pumps just mentioned. And other devices are brought to the bedside to address chronic or acute conditions; these would include hemodialysis machines and defibrillators. Finally, voluminous data records are being created for each patient from the output of all these machines. Vital-signs measurements, along with any changes in the therapeutic regimen, need to be captured in the patient record and often communicated elsewhere, such as to the pharmacy.

Surely by now one might assume these monitoring devices supply the information they gather directly to the systems that chart patient data. Unfortunately, not to the extent one might imagine. The standardization of communication processes that has led to the explosion of telecommunications products in the consumer area has yet to take hold in the world of clinical medicine. This lack of connectivity leaves open to question the accuracy and completeness of a patient record created by harried clinicians whose attention to data entry tasks diverts their attention from patients. And without electronic capture of data and events associated with an episode of care, trending and other sophisticated data analyses are effectively impossible.

The scope of the problem just outlined is not limited to the ICU. It extends to all locations where healthcare is delivered, even to the home.

Application of already existing data-communication standards and models might seem to be all that is required to address the situation. To some extent this has occurred, but it has not been enough. Information technology (IT) standards within the commercial application domain (e.g., IEEE 802.x standards) are inadequate to fully address the needs of the clinical IT domain, particularly at the patient bedside.

As implied, the solution to the problem of transferring data from one medical device to another or to a clinical information system is to develop an interface to support

that intercommunication, preferably by leveraging existing standard communications technologies. Point-of-care medical device communications (POC MDC) standards now under development could answer the need for plug-and-play medical device interoperability. This article defines and describes these standards and reports on their status as works in progress.

The Interoperability Problem

The need is for intercommunication among medical devices and clinical information systems. Indeed, this has been accomplished with a number of medical products. Infusion pumps and ventilators commonly have RS-232 ports, and these devices can communicate with many physiological monitoring instruments. Products to link medical equipment and personal communication devices exist as well. However, virtually all of these are specialized applications—custom interfaces unique to the two devices being linked. The fact that an infusion pump from Company A can communicate with a patient monitor from Company B does not guarantee that Company A's pump can communicate with the same type of monitor from Company C.

Interfacing two devices with "standard" RS-232 ports does not ensure communication, because there are many different ways to send data over that serial interface. Matching the connectors and pins can be problematic, as is establishing a handshake. Moreover, medical device design is not perfected simply because data can be sent from one device to another; the devices must be able to understand the format and content of the messages they communicate to each other; that is, they must speak the same language, both grammatically and semantically.

In a clinical setting like the ICU, devices are brought to the bedside when needed and set up by clinicians whose focus is on the patient, not the technology. Frequent connection and disconnection is normal. Clinicians do not have time to run configuration or setup programs; rather, they expect plug-and-play functionality. Such devices ought to be designed so that they automatically integrate and interoperate with the bedside system.

Some products do attempt to provide this level of integration, as noted above, but within a system that is to some extent proprietary or "closed." Although a hospital may try to standardize its equipment as much as possible, it is not unusual for more than one model of infusion pump, ventilator, patient monitor, or other device to be used. The cyclic replacement of equipment with new makes and models militates against achieving standardization.

The core of the so-called plug-and-play interoperability problem is this: In the absence of a communications standard that extends from the physical device connection through the application-language level, every interface between a medical device and any device or system with which it is to communicate must, at a minimum, be examined to determine what physical and logical interfaces must be developed to effect communication. The expenditure of resources will be required in virtually every case to develop and maintain the needed interface and to support the required system integration.

This problem is not unique to medical devices; it affects all healthcare IT. Although many physical-layer issues have been resolved, work on interoperability at levels

closer to the application is still receiving considerable attention. Standards groups such as Health Level Seven (HL7) and its Clinical Context Object Working Group (CCOW) are focused on resolving the problem for large-scale systems. The problem is approached with respect. Its scope and cost for a healthcare enterprise are not insignificant. For any two systems intended to interoperate, an interface must be built. Add a third system with communications hardware or software protocols that differ from those of the others, and two interfaces must be built and integrated. Consider the implications for a market with thousands of devices that may need to communicate with one another.

Every dollar and every minute devoted to developing a communications interface is time and money not committed to a healthcare-related application. Many systems-level problems affect the safety of everyone involved in healthcare delivery. Such systems are often also expensive. It is unfortunate that engineering talent must be focused on the development and maintenance of specialized equipment interfaces rather than on solving problems that result in undesirable risks and costs.

The Medical Information Bus

To address the medical device plug-and-play interoperability problem, a single communications standard is needed. Software engineers designing medical equipment could use such a standard to implement external interfaces once for all models. POC MDC, or MIB (medical information bus), standards are poised to fill this need. MIB is the common name for a series of standards published or under development by the Institute of Electrical and Electronics Engineers as the IEEE 1073 standard for medical device communications. Intended to bring a wide range of medical devices under its purview, these standards aim to encompass transparent plug-and-play interoperability, ease of reconfiguration, and ease of use. Other important requirements that have been identified include the following.

Safety. Medical devices must adhere to all relevant patient and user safety regulations. This includes their external communications interfaces.

Unambiguous Association. A medical device must be able to be related to a specific patient.

Unambiguous Device Identification. Throughout the healthcare enterprise, the device must be uniquely identifiable. It could even be argued that, given wide-area network (WAN) technologies, this identification must be maintained for all devices worldwide.

Wide Range of Topologies. Medical device networks could be confined to a single room or could extend over many beds or care units. In telemedicine applications, a device's data could be viewed, and functions controlled, from a distant site.

Reliability. Communications should be robust, and a single-point failure should not disrupt an entire network.

Cost to the User (Commercial Viability). Although not specifically addressed in any of the IEEE 1073 standards, the incremental cost of adding MIB to a device must be reasonable and make good business sense.

Off-the-Shelf Technologies. As often as possible, standards-based technologies should be used that are readily supported in the marketplace by software and hardware components.

Bandwidth (Communication and Processor). Bandwidth adequate to support the data rates associated with the family of devices using a link must be provided. In addition, because many medical devices use a single embedded processor to support all functionality, the communications bandwidth could be shared with the other processing functions of the device.

Power Consumption. Power for the communications subsystem will be supplied from the source that supports the primary function of the device, which is often a battery. The operation of the communications subsystem cannot degrade device performance. In fact, for very small devices, the interface could provide all power.

Internetworking. The interfaces should support the communication of medical device data across a variety of network configurations, including standard line extenders, protocol converters, bridges, routers, gateways, and terminal concentrators. An example is the addition of a modem pair to the device-host link to support remote monitoring. This may also include provision of power by either side of a link to power small devices.

International Support. Many devices, or at least device companies, support an international market. The communication standard should be the same in the domestic U.S. market as in European, Far Eastern, and other markets.

Legacy Devices. Where possible, the standard should lend itself to retrofitting in existing (legacy) medical device designs.

LAN Access. Although a medical device may have a simple point-to-point connection at the bedside, the application that is interfacing directly with the device can be located on a local-area network (LAN) within the facility. Also, a device might want to search for and utilize services that are provided by entities connected to a LAN.

Time Synchronization. Systems must be able to synchronize data acquired from several devices, especially for the real-time display of multiple waveforms.

HL7 Interoperability. No matter where data begin, at some point they almost invariably must be translated to HL7 (for example, via a gateway process) for consumption by applications such as archival repositories, pharmacy systems, and ADT (admit-discharge-transfer) systems.

Security (and HIPAA). Once device data are routed beyond the immediate host connection (for example, via a LAN to a remote application), security must be maintained to ensure privacy and source authentication. This is especially true given the requirements mandated by the Health Insurance Portability and Accountability Act of 1996 (HIPAA).

Remote Control. The standard interface should be robust enough both to support remote-control applications and to provide a standard control interface, regardless of the type of device.

Alarm Management. A protocol for reliably annunciating device alert conditions in a timely fashion should be supported.

Scalability. Some medical devices are fairly complex (e.g., ventilators), whereas others are relatively simple (e.g., pulse-oximeters). Host systems also vary in complexity. The interface should be scalable, both statically and dynamically, to support this variation in complexity.

The IEEE 1073 MIB standards for medical device communication, along with their international counterparts, address each of these requirements and more.

In order to satisfy these requirements, early MIB developers conceived a branching LAN topology. Each link in an MIB implementation is formed via the connections between a host and a device, termed a bedside com-munication controller (BCC) and adevice communication controller (DCC), respectively. Each BCC communicates with one or more DCCs, but each DCC must communicate with only one BCC. A given medical device can function as both a BCC and a DCC. For instance, a bedside monitor can be a BCC connected to ventilator and infusion pump DCCs, while at the same time it can be a DCC connected to a clinical information system acting as a BCC.

The heart of MIB is therefore the BCC-DCC interface, and the specification of this interface and its realizations is the work of the IEEE 1073 General Committee and related organizations. Like most modern communications protocols, MIB is generally patterned after the International Organization for Standardization's Open Systems Interconnection (OSI-ISO) seven-layer communications model.[1] That model was created to foster interoperability between communicating systems by isolating functional layers and defining their abstract capabilities and the services relating adjacent levels. The four so-called "lower" OSI layers are the (1) physical, (2) data link, (3) network, and (4) transport layers. Layers 5, 6, and 7—the session, presentation, and application layers—are known as "upper" layers. Figure 1 illustrates the logical interface between two MIB-connected systems, a Manager (typically a host/BCC) and an Agent (typically a device/DCC). The figure provides an overview of the logical interface and the component elements of an MIB system.

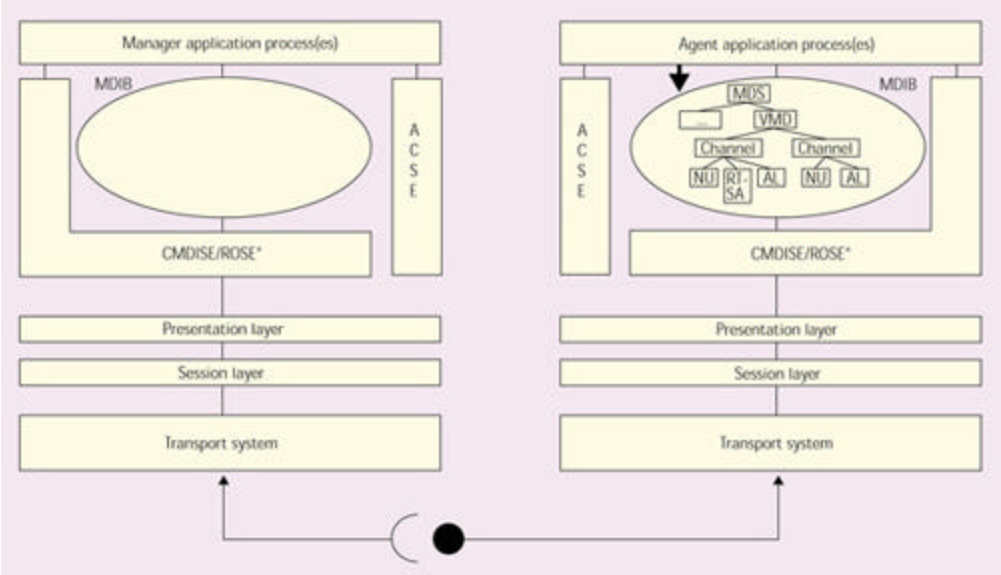

Figure 1. The logical interface between two MIB-connected systems. This schematic is taken from the CEN 13734 standard for Vital Signs Information Representation (VSIR). A more complete treatment of this example is found in VSIR Annex C, "Communicating Systems Example."

Transport System. Layers 1–4, the "lower" layers, constitute the transport system, which provides reliable transport of data across different media.

Session Layer. This functional layer includes services for connection and data transfer (e.g., session connect, session accept, and session data transfer).

Presentation Layer. This layer holds services for negotiating abstract syntax, such as Medical Device Data Language (MDDL) over CMDISE ASN.1 (see below), and transfer syntax, which are basic encoding rules (BER) or optimized medical device encoding rules (MDER). MDERs are abstract message definitions that include primitive data types such as FLOAT (floating-point numeric) or 32-bit integer, and the way they are encoded as bits and bytes for communication over the transport.

ACSE. The association control service element (ISO/IEC 8650) provides services used initially to establish an association between two communicating entities, including association request and response, association release, association abort, and others.

ROSE*. The remote operation service element (ISO/IEC 9072-2) provides basic services for performing operations across a connection, including remote operation invoke, result, error, and reject. The asterisk indicates that an optimized version of the protocol is employed for medical device communications.

CMDISE. This is the common medical device information service element, based on CMIP (the common management information protocol; ISO/IEC 9596-1). It provides basic services for managed objects, including the performance of GET, SET, CREATE, DELETE, ACTION, and EVENT REPORT functions. These services, invoked using ROSE primitives, represent the basic means for interacting with the medical data information base (MDIB).

MDIB. The medical data information base supplies an abstract object-oriented data model representing the information and services provided by the medical device. The data originate in the device agent (the right side in Figure 1) and are replicated during connection on the Manager side of the system. Objects include the medical device system (MDS), virtual medical device (VMD), channels, numerics, real-time sample arrays, alerts, and others.

Application Processes. This layer represents the core software on both the host (BCC) and device (DCC) sides of the connection that either creates or consumes the information that is sent across the link.

To someone unfamiliar with standardized communications models and technologies, this arrangement may appear to be a much bigger hammer than is needed for the job. (A common question is, "Can't we just send an ASCII string with 'RATE=125mL' across the link?") But a level-by-level examination reveals that it solves aspects of the communication puzzle in a standardized and straightforward manner that enables bedside plug-and-play interoperability to be achieved.

The medical device communications problem has three principal parts: lower-layer, or transport, services; upper-layer application profiles; and upper-layer semantic, or device-specific, object data models. For each part, the technology that best fills the needs of a given medical device and host system can be selected without having a major effect on the other two parts. The specific combination of all three technologies is determined (or discovered) during the initial configuration of the communications link (transport connection, association, and data model discovery)—all without clinician intervention.

The MIB Family of Standards

The IEEE 1073 standards set generally reflects the tripartite structure of the medical device intercommunication challenge. It consists of a base standard, which provides an overview and framework for the set, and the following focused standards, which either have already been published or are scheduled to complete ballot in 2001.

1073.1: Medical Device Data Language. The MDDL standard covers nomenclature (the set of unique 16-bit codes used to name elements in the data model), generic object patterns used for different applications (e.g., an alarm pattern), and device-specific standards. The designations of the parts of this standard are:

- 1073.1.1: MDDL—Nomenclature.
- 1073.1.2: MDDL—Generalized Device
- 1073.1.3: MDDL—Specialized Device (with 1073.1.3.x sections pertaining to specific device types).

1073.2: Medical Device Application Profile (MDAP). This standard defines the set of services that will be used to communicate MDDL information between the DCC and BCC systems. Sections under this rubric cover the basic encoding and abstract syntax for messages used by ACSE, ROSE, and CMDISE; event-report messages, or protocol data units (PDUs), sent by devices to the host; and services used when the host "requests" information from a device.

- 1073.2.0: MDAP—Base Standard, which provides PDU definitions for ACSE, ROSE, and CMDISE services, as well as medical device encoding rules and a specification for the Medical Device Numeric Format (MDNF) used for communication of real numbers and other primitives.
- 1073.2.1: MDAP—Baseline Profile, defining a mainly event-driven set of services by which the data model, or "containment tree," is sent by the device (DCC) during link configuration. Subsequent information is sent primarily as event reports automatically generated by the device when there is a change in operational status or new data are available.
- 1073.2.2: MDAP—Polling Mode Profile, defining a set of services that allow a host system to "poll," or explicitly request, all data to be sent from the device; that is, the device sends data only when the host (BCC) has asked for it.
- 1073.3 and 1073.4: Transport Profiles. These standards cover the lower-layer functions, namely, physical connection through transport.
- 1073.3.1: Transport Profile—Connection Mode, the original MIB transport standard that addressed OSI layers 2–4, though the transport and network layers were null definitions; it is mostly concerned with the data-link layer.
- 1073.3.2: Transport Profile—IrDA Based—Cable Connected, a more recent standard based on the infrared standard published by the Infrared Data Association (IrDA) and used in most peripherals with infrared interfaces. However, the standard defines a cable that uses RJ-45 connectors at either end and has RS-232 signaling levels. It was created as a key means of facilitating incorporation of legacy medical devices into an MIB network, taking advantage of the fact that many devices being used today have RS-232 ports. The maximum transmission speed is 115K baud.
- 1073.3.3: Transport Profile—IrDA Based—Infrared Wireless, a standard in the process of being finalized that is based on work done on the 1073.3.2 standard but that uses infrared rather than a cable; the maximum speed is 4 Mbyte, "fast infrared." This standard is also based on work done by the LAN Access Point team of the Point of Care Connectivity Industry Consortium. It will contain an annex illustrating how medical devices can use LAN access points to connect medical devices, especially point-of-care test devices, such as glucometers, to remote applications using a TCP/IP (Transmission Control Protocol/Internet Protocol) link.
- 1073.4.1: Physical Layer Interface—Cable Connected, the original MIB physical connection standard that includes a unique connector and can run at speeds up to 1 Mbyte.

The use of RS-232 in 1073.3.2 differs from the limited applicability described in the discussion of the interoperability problem above in that it is considered within the context of a complete seven-layer communications model. The IrDA standards (IrLAP, IrLMP, and TinyTP) were developed for infrared communications, but their protocol stack meets IEEE 1073 requirements and is available in off-the-shelf tool kits from a number of vendors. Not having to develop the stack frees precious development resources for application-oriented concerns.

A fourth general standards area, internetworking (IEEE 1073.5), is being revived after lying dormant for several years. It would address the broader questions of upper-layer communications across a LAN (e.g., TCP/IP), gateways to protocols such as HL7, and the use of bridges, routers, relays, and other internetworking devices. Additional lower layers are also envisioned for the future, including radio-frequency wireless and TCP/IP. More information about IEEE 1073 is available on-line at http://grouper.ieee.org/groups/mib/.

Making the Connection

To ensure orderly system behavior, MIB describes the fundamental finite-state-machine model for the life cycle of a BCC-DCC interaction. Figure 2 illustrates the state model for a device interface. Specific application profiles (i.e., polling-mode or baseline) are to use this as a generic-state machine, but they may define specific requirements or assumptions for the individual state transitions. After a connection is made at the transport level (indicated in the figure by the "connection" event), the DCC proceeds to associate with the managing BCC system and configure the link. Once configuration has been completed, the communication enters the normal operating state in which, in accordance with the profile that is active, data may be exchanged between the two systems.

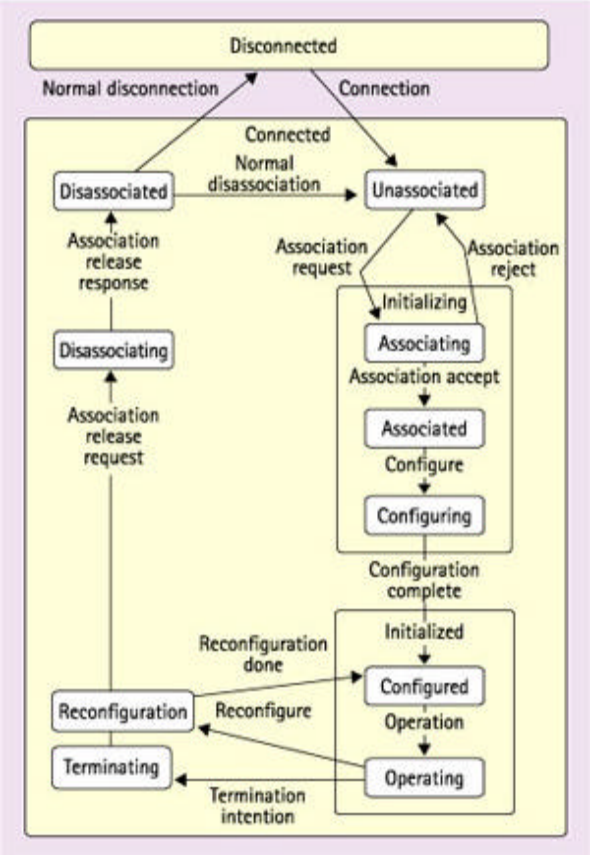

Figure 2. A generic device MIB communications-state machine. The diagram is taken from the standards CEN 13735, "Health Informatics—Interoperability of Patient-Connected Medical Devices," and IEEE 1073.2.0, "Medical Device Communications—Medical Device Application Profiles (MDAP)—Base Standard, Annex D: Dynamic Model."

If the device is reconfigured—for example, if a new plug-in module is added—it can transition through the reconfiguration state, in which the Manager is notified of the changes in the Agent's MDIB data model, and then cycle back to the operating state. The final state is Figure 3 illustrates a sequence model of the interactions between an Agent (DCC) system and a Manager (BCC) system. Here, once the Manager transport layer indicates that a connection has been made, the Manager appli-cation, using ACSE PDUs, initiates the association-establishment process, which results on the Agent side in the association-request event being generated. Association being accomplished, the Agent notifies the Manager that the MDS object has been created. This MDS-create-notification event report includes static information about the device's manufacturer, its serial number, and other configuration data.

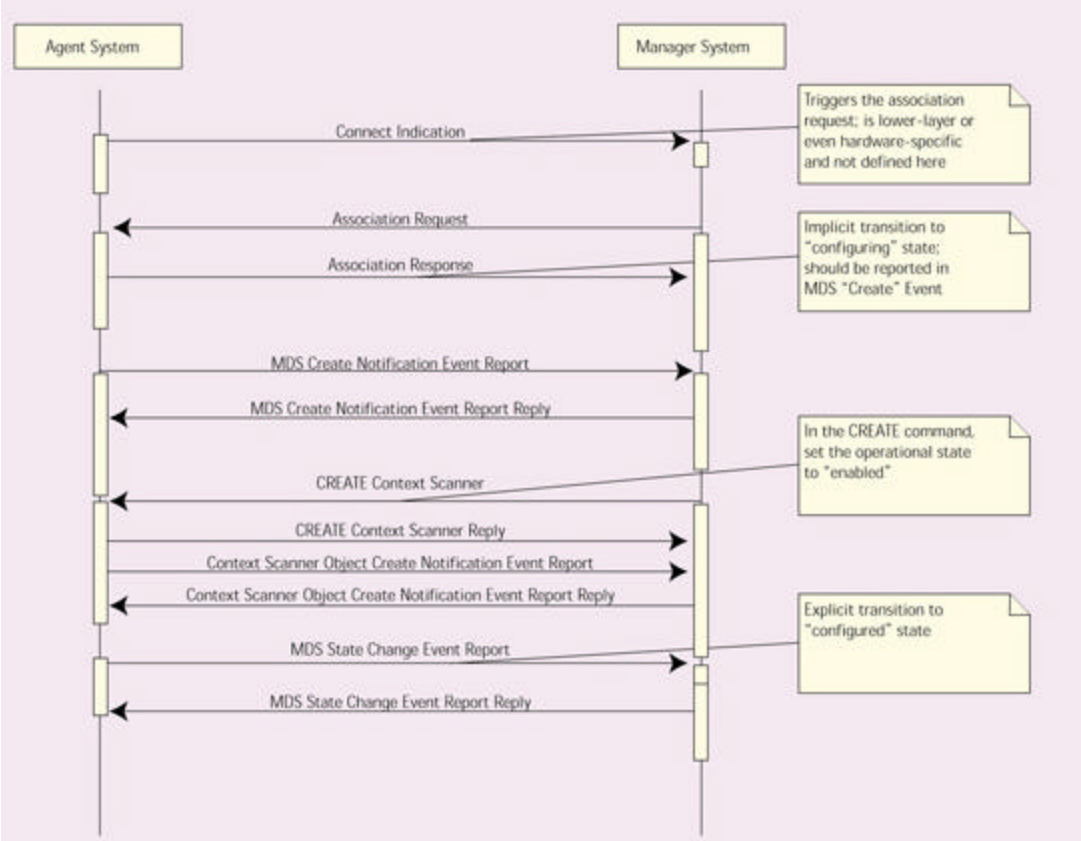

Figure 3. A baseline application profile system interaction.

At this point, the Manager can create a context scanner within the device's MDIB. A scanner is a tool that collects information of various kinds from the device's MDIB and sends it to the Manager in event-report messages. A periodic scanner will examine a set list of data items in the MDIB (for example, in an infusion pump, this list might include the parameters "volume infused" and "volume to be infused"), and send an update at regular intervals of every few seconds.

In the infusion-pump example, a context scanner is used to report the object-model containment tree to the Manager system. This way, the Manager can "discover" the data that are supported by a given device. Because the MDIB contains a finite set of object types (MDS, VMD, channel, numeric, alert, battery, etc.), a Manager does not need to know what an infusion device looks like, it can simply process the containment tree retrieved from the context scanner and configure itself accordingly. Once the containment tree has been sent to the Manager system and the Agent has received a confirmation reply, the MDS object indicates that it has entered the configured state. Then, as shown in the state model in Figure 2, automatically passes to the operating state, ready to begin regular data communications.

This is a simple illustration of an MIB connection's dynamic behavior. Additional objects are provided for patient information, batteries, event logs, various sample arrays, alarm management and reporting, and even remote control. All of these objects use either the polling-mode or baseline application profiles. They use the same basic set of ACSE, ROSE, and CMDISE services.

Harmonization with Other Standards

IEEE 1073 standards have not been created in a vacuum. A number of related standards were developed by the European Committee for Standardization (CEN) in

close harmonization with MIB efforts in the United States. Working Group 4 of CEN Technical Committee 251 (some of whose members are also on the IEEE 1073 committee) has drafted ENV 13734, "Health Informatics—Vital Signs Information Representation," a standard that defines the basic object-oriented data model for the MDIB (referenced in IEEE 1073.1.2).

The working group has also drafted a version of the nomenclature that is highly harmonized with the IEEE 1073.1.1.1 draft standard. Additionally, the ENV 13735 standard, "Health Informatics—Interoperability of Patient-Connected Medical Devices," defines the two basic application profiles (referenced in IEEE 1073.2.1 and 2.2) along with a number of optional packages, such as support for the patient-demographics object. This standard also relies heavily on IEEE 1073.2.0 for encoding and for PDU definitions of ACSE, ROSE, and CMDISE.

Both the CEN and IEEE standards are fast-tracked to become ISO standards through the efforts of ISO Technical Committee 215, Work Group 2 on healthcare informatics messaging and communications. When that happens—when medical device standards heretofore informally harmonized become part of a single set of international documents—a manufacturer can build to a single basic communications protocol definition and expect to have its devices work worldwide.

Becoming a Reality

The first public demonstration of a full implementation of IEEE 1073 using the IrDA RS-232 lower layers took place in Boston in February 1999. Infusion pumps provided by Alaris Corp. (San Diego) communicated interchangeably with a GE-Marquette patient monitor and Hewlett-Packard (now Agilent) device interfacing system.

On the same day as that demonstration, IEEE announced the formation within its Industry Standards and Technology Organization (IEEE-ISTO) of the Medical Device Communications Industry Group (MDCIG). MDCIG serves as an industry forum to support activities associated with the completion of MIB standards, prototyping of systems based on those standards, and maintenance of the technologies (for example, updating the nomenclature). MDCIG includes Abbott Laboratories, Agilent, Baxter, Caducian, Marquette (now GEMS-IT), and Siemens. The group has made MIB software available to developers at no charge. It has also published a white paper describing how MIB-enabled applications can be used to address systems-level healthcare safety problems highlighted in a recent Institute of Medicine report.[2] To learn more about MDCIG and its activities, go to http://www.mdcig.org.

A second demonstration project, involving infusion pumps, ventilators, monitors, and host applications from different manufacturers, is scheduled to be ready in the summer of 2001. A third demonstration—based on telemedicine applications—is also in the works, planned for completion in mid-2002. This effort will focus more on internetworking/WAN and remote-control features.

MIB has been in development since the mid-1980s. With the passage of years, some people have concluded that it will never become a reality. But in view of the advances of recent years—the advent of new lower-layer technologies that are used throughout the computer industry, the international scope of the standards creation effort, and the significant participation of many major medical device vendors in the IEEE-ISTO MDCIG—there is little doubt that MIB is rapidly approaching fulfillment.

Conclusion

The domain of POC MDC and MIB is a territory that encompasses connectors and stacks, patient safety, object-oriented models and finite-state machines, topology requirements, and grammatical syntax and semantics. To realize safe, robust, plug-and-play interoperable medical device communications, all of the issues within the problem space must be addressed.

Much progress has been made in constructing the first building-block standards that provide a basis with which to construct intercommunication systems. It is up to the medical device manufacturing community and its customers to determine whether the value of interoperability and the benefits of standards-based communications make it worth the effort to bring realizations of MIB to market.

REFERENCES

1. *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model,* ISO/IEC 7498-1 (Geneva: International Organization for Standardization, 1994).
2. Committee on Quality of Health Care in America, Institute of Medicine, *To Err Is Human: Building a Safer Health System* (Washington, DC: National Academy Press, 2000).