Standardized acquisition of bedside data: The IEEE P1073 medical information bus

M. Michael Shabot
The Department of Surgery, Cedars-Sinai Medical Center, Los Angeles, California

Accepted 27 September 1989

Key words: medical, information, bus, network, computer

Abstract

The absence of standards for medical device communications has stymied the acceptance and success of automated clinical data management systems. Even devices with simple RS-232 data output ports require special interfacing hardware and software. Due to the number and variety of medical devices available, each with their own peculiar data output configuration, it has been impractical to interface with most of them. Limited by manual data entry, most computerized patient data management systems have failed to deliver the productivity gains their users expected.

The forthcoming IEEE P1073 Medical Information Bus (MIB) Standard promises to correct this situation with a single powerful bedside device interface method. The MIB will provide specifications for all hardware and software necessary for medical data communications. The MIB handles the need for automatic recognition of new devices placed at a bedside, automatic reconfiguration of the network, binding of a device to a particular patient’s bedside and many other issues unique to the medical data communications environment. The MIB is expected to undergo formal IEEE balloting in 1990 and promises to open a new era in data management for clinical patient care.

Introduction

Imagine the following scenario: your team has just completed the implementation of a successful computerized Patient Data Management System (PDMS) in your busy Intensive Care Unit. A new instrument, an electronic urimeter which continuously measures urine output and core body temperature (Bard Urotrack Plus, C.R. Bard Company, Murray Hill, NJ), is introduced into your ICU. The device is interesting because it has an RS-232 data output port which reports all device measurements every second. Furthermore, the data output port appears to be as straightforward as possible, because it is the ‘simplex’ (send only) implementation of RS-232 and used only two wires, ‘data out’ and ‘ground’. It seems both easy and attractive to interface this device to the PDMS in order to automatically capture and display urine output and core body temperature.

The scenario above occurred at Cedars-Sinai Medical Center, but the reality of the implementation was much more complex than originally envisioned. As frequently occurs, the RS-232 ports on the PDMS computer (Hewlett-Packard 78709 PDMS, Hewlett-Packard Co., Waltham, MA) were configured differently from the urimeter’s RS-232 output ports. A protocol conversion device was required to resolve the incompatibility. In addition, one computer port had to be fanned out to
20 bedside urimeters using two multichannel controllers. Extensive PDMS software had to be written to switch the controller ports and acquire urimeter data. Eventually the interface required the purchase of several thousand dollars worth of hardware and months of software development [1].

The results were gratifying, however. Since our ICU delivers 5,600 patient days of ICU care per year, we conservatively estimated that the interface could eliminate up to 134,400 manual urine output measurements (at a savings of 20 seconds per measurement), 33,600 rectal temperature measurements (saving 2.5 minutes per measurement) and 134,400 instances of manual results charting (saving at least 10 seconds per instance). Potential nurse time savings were calculated at over 2,500 hours per year, or 27 minutes per patient day [1].

A few months later, the Medical Center installed bedside pulse oximeters (Ohmeda Biox 3700, Ohmeda Co., Boulder, CO). This device also has an RS-232 data output port and again it seemed desirable to automatically capture real time saturation values and alarms for the PDMS. However, the configuration of the RS-232 on the pulse oximeter was different from the urimeter and the PDMS computer. As a result, all new software and a separate RS-232 data communications network had to be created for the pulse oximeter. Only the general schema for polling bedside devices and storing results into the PDMS data base could be transferred from the urimeter experience.

The utility of interfacing multiple bedside devices to a data management system cannot be denied. Figure 1 shows a Cedars-Sinai PDMS real time status display screen which combines inputs from the bedside urimeter, oximeter and physiologic monitor. To this point virtually all the pioneering efforts in medical data management have involved the construction of bedside data links [2, 3, 4]. However, it remains impractical for most medical centers to develop this hardware and software [5].

**Problem statement**

A concise statement of the problem is that *absence of interface standards for bedside medical devices has precluded the connection of most bedside devices to patient monitoring computer systems and alarm networks*. In addition, the absence of standards has markedly limited the success of comprehensive clinical computer systems, because most bedside data cannot be captured automatically. Although virtually all new medical devices have a data output port, these ports adhere to different versions of various hardware and software standards. It is uneconomical for hospitals and medical computer system vendors to provide different interfaces to the myriad of bedside devices currently available.

**Pilot standards efforts**

A pilot Medical Information Bus has been developed by Gardner, Hawley and colleagues at the LDS Hospital in Salt Lake City, Utah [6, 7, 8]. Infusion pumps, ventilators, pulse oximeters and other bedside instruments now transmit data to host computers in the ICU. Although the precise design of the LDS MIB differs from the IEEE P1073 Draft Standard, valuable lessons have been learned and problems solved with the LDS experience. For instance, the importance of implementing a ‘filter’ function to control device reporting was an LDS insight, because without such a filter, minor variations in physiologic measurements were found to flood the network with trivial data. The IEEE P1073 Medical Device Data Language (described below) now specifies a variety of filter functions.

**The IEEE P1073 Medical Information Bus Committee**

In 1982, a small group of individuals met in the boardroom of Phoenix Baptist Hospital, Phoenix, Arizona, to form a Medical Information Bus Committee. The objective of the Committee was to develop a hardware and software communications standard for medical devices which would permit ‘plug and play’ connection of bedside devices to