Emerging medical applications require networked coordination between medical devices. However, most of the medical devices in use today do not support wireless communication or network connectivity. Currently, hospitals interested in coordinated medical care would have to replace existing devices with expensive new devices. We believe that existing medical devices can be extended to support interoperable network connectivity. We demonstrate the Medical Device Dongle (MDD), an open-source platform that enables such extensions to current medical devices. The MDD can attach to any device that has a data output interface (RS-232 or USB) and enables it to connect wirelessly and in an interoperable manner for various applications. We show how multiple medical devices, including pulse oximeters and infusion pumps, can be connected and controlled together using an open-source platform, standards-based connectivity protocols, and model-driven software. The demo setup consists of medical devices attached to an MDD Agent, an MDD Manager device, and a mobile phone running monitoring applications. The MDD components can communicate over Bluetooth, WiFi and Ethernet.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Wireless Health ’11, October 10 - 13, 2011, San Diego, USA
Copyright 2011 ACM 978-1-4503-0982-0 ...$10.00.
2. MDD OVERVIEW

Applications of interest typical require that medical devices connected to a patient must interact with a central point (usually called the supervisor). This supervisor manages the coordination between the devices and runs applications that use these connected medical devices. There are, therefore, two versions of the MDD, designed to fit into this multiple-devices-single-manager architecture: one for devices connected to the patient (the agent MDD); and another for the supervisor (the manager MDD). The MDD can be implemented physically as a peripheral that connects to a device or logically as software components running on the device (with access to the device’s network interface). The main part of the MDD is this collection of software components, which we call MDDWare. This demo shows the agent MDD implemented as a physical peripheral and the manager MDD implemented as a physical device in one case and logical software components in another.

Rather than develop our own interoperability protocol, we chose the MDD on the IEEE 11073 protocol. The MDD is also designed to support other medical device and interoperability protocols, especially those geared towards maintaining patient safety. We chose 11073 because it is an IEEE standard and supports the multiple-devices-single-manager model. The network architecture is based on the older point of care (PoC) standard, while the connectivity and communication protocol is based on the personal health devices (PHD) protocol. We chose PHD for the main protocol because it has better and more recent documentation than the previous PoC standard. It is important to note that we did not implement the full standard, but only those parts we deemed sufficient for medical device connectivity and communication. In particular, we implemented: 1) manager and agent finite-state machines for maintaining connectivity and providing GET and SET services for message exchange and command execution; 2) the Medical Device System (MDS) object from the domain information model (DIM) for device description; 3) the Medical Device Encoding Rules (MDER) from the communication model for encoding messages. We call this implementation 11073-MDD. The MDD turns any medical device with a data output interface (RS-232 or USB) into an 11073-MDD-compliant device.

3. DEMO SCENARIO

Supposing the patient room is an Intensive Care Unit (ICU), medical devices are connected to the patient to measure vital signs such as pulse oximetry (SpO₂), heart rate or to administer treatment such as insulin infusion. Figure 1 shows the scenario of our demo, which consists of three main parts: 1) Multiple medical devices: the data source for the manager; 2) Agent MDD: the intermediary between medical devices and the MDD manager; and 3) Manager MDD: An embedded device or smartphone. The relevance of MDD in such a scenario is highlighted as follows:

**Agent MDD.** The agent MDD behaves like a translator between the manager MDD and the medical device. It translates any data from the device into IEEE 11073-PHD compatible formats and transmits the data to the MDD Manager over a network interface.

**Manager MDD.** The medical device manager is in charge of coordination of different devices. It monitors the connection of the supervisor to agent devices and is responsible for sending and receiving messages on behalf of medical applications such as a smart alarm [2] running on the supervisor directly connected to the manager as shown in Figure 1.

Figure 1: Device Connectivity Architecture

4. REFERENCES


