A Programmable Service Architecture for Mobile Medical Care

Rajiv Chakravorty
University of Cambridge Computer Laboratory
JJ Thomson Avenue, Cambridge CB3 0FD UK
Rajiv.Chakravorty@cl.cam.ac.uk

Abstract

This paper introduces MobiCare – a novel service architecture that enables a wide range of health-related services for efficient and mobile patient care. These services include: (1) health-related services in medical devices and sensors to remotely install, self-activate, reconfigure or even self-repair with new health services and applications, (2) secure and reliable dynamic software upgrade or update services applied to the native code of the clinical device, and, (3) remote registration and (re)configuration of body sensors as well as remote health-data services such as patient health report downloads and diagnosis data uploads with provider servers. Collectively these services address a range of patient medical monitoring needs by accelerating deployment of new health-related services, thus reducing medical costs and improving the quality of patient care. We are currently implementing a proof-of-concept prototype. Early experiences with MobiCare do show that it has the potential to become a feasible and a useful infrastructure paradigm for the next generation healthcare.

1 Introduction

A significant proportion of the human population suffer from various chronic ailments. Under the existing healthcare systems, for example, the fatality rate in the US from heart failures itself is more than 42%, many of which are due to the delays incurred in initiating medical intervention.

This paper introduces MobiCare – a wide-area mobile patient monitoring system that enables continuous and timely monitoring of patients thereby enhancing quality of care for patients and potentially saving many lives. MobiCare consists of three important building blocks (Figure 1): a body sensor network (BSN) consisting of wearable sensors and actuators that are inter-connected using the wireless medium; a BSN Manager (also called the MobiCare client device) that connects the BSN to an 'always-on' communication wide-area interface, e.g., a GPRS/UMTS cellular link; and backend infrastructure support (servers) at healthcare providers that provide necessary healthcare services to patients.

The main goal in MobiCare is to define a programmable remote patient monitoring infrastructure that exploits the recent advances

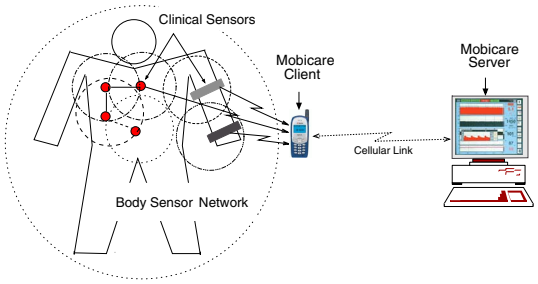

Figure 1. Medical services with MobiCare.

in wireless cellular and clinical sensor systems. By leveraging these advances, MobiCare can offer potential health benefits: (1) continuous monitoring for chronically-ill patients, (2) better quality care and feedback for patients, (3) increased medical capacity, and, (4) reduced medical costs for patient care.

Enablers of MobiCare

A number of recent innovations enables the efficient design of the MobiCare architecture. Advances in medical sensors today enable efficient, remote monitoring of patients. For example, sensors to measure ECG are now commercially available from Numed [5] and Health Frontier [1]. Agilent, Philips and Nelcore produce handheld pulse oximeters for noninvasive monitoring of blood oxygen saturation and pulse (SpO_2). Other sensors from Nonin [4] and Linde AG [3] use wireless connectivity (Bluetooth-based) to provide remote monitoring of vital body signs. Vendors like omron (`www.omron.com`) produce a range of portable wrist devices. The CodeBlue project at Harvard has also developed (using the Berkeley MICA2 mote) a low-power, low-frequency, wireless pulse oximetry and ECG sensor for patients [10]. With clinical sensing technologies advancing at a much faster rate one can expect a range of such energy-efficient wireless medical sensors devices to become available. MobiCare exploits such sensors to construct the BSN that monitor the patient's health non-invasively to gather vital health data, e.g., heart condition, blood pressure, serum glucose level, temperature, oxygen saturation (0_2). These sensors in BSN self-organize and connect to a wireless interface to communicate such data to the BSN Manager and ultimately to the remote back-end servers.

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE
IEEE COMPUTER SOCIETY

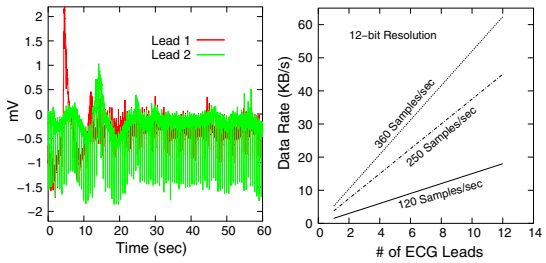

Figure 2. Shows (a) ECG signals, and, (b) corresponding ECG data-rate requirements.

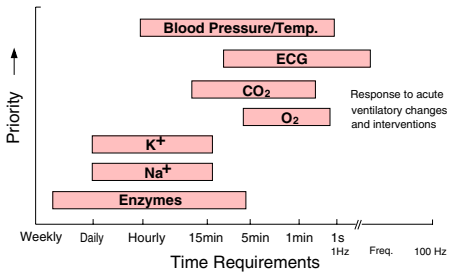

Figure 3. Shows typical priority and time requirements in clinical measurements.

MobiCare exploits the continued increase in coverage and bandwidth of cellular wireless networks being deployed worldwide to build the 'always-on' wide-area interface of the BSN. For instance, the newly deployed UMTS 3G network in Europe and CDMA 2000 EvDO network in the US can provide for data-rates that are much higher than offered by conventional fixed-wire dial-up modems. Such mobile networks therefore open up possibility for patients to be continuously monitored and their vital health data efficiently transported from the BSN to back-end servers, thereby enhancing the timeliness and quality of medical care.

Design Goals in MobiCare

To identify the key design attributes of a mobile health care system such as MobiCare, we investigated the monitoring requirements for patients. Figure 2 illustrates an example of monitored ECG data from a heart patient and the corresponding data-rate requirements when the number of such ECG leads are varied. Figure 3 also summarizes the typical time requirements and relative priority of some vital body signs including blood pressure, blood gases (O_2 and CO_2), heart condition (ECG), and enzymes. Thus design of a mobile healthcare system should consider the medical requirements to derive the design goals. We derive the following important observations.

First, vital body signs have different *time requirements* in patient monitoring (Figure 3). Such monitoring may be needed periodically or may involved on-demand continuous monitoring. Time requirements of such monitoring vary significantly – from few minutes to several hours – depending on the condition of the patient and the severity of the ailment. Flexible and remote configuration of sensors is therefore crucial for effective mobile medical monitoring. Second, clinical sensors that address different monitoring requirements of patients can potentially be built by different vendors and may use different wireless protocols and technology, each operating in a different part of the wireless spectrum. Such uncoordinated clinical sensor design (as exists today) can make integration and self-organization of such body sensors exceedingly difficult to achieve. In order for such sensors to function together in a single BSN, remote adaptation and reconfiguration should be an integral requirement of the architecture. Finally, due to its time-critical nature, reliability, security and timeliness of data delivery from the BSN to the back-end servers is crucial.

MobiCare achieves these goals with the following features:

Devices reconfigurability. A mobile health care system should be able to dynamically integrate, organize and configure new body sensors based on the needs and the requirements of patients and health providers. MobiCare fulfills these design goals with secure and dynamic code update functionality that is implemented as part of each MobiCare client and sensor device. Note that secure reliable medical sensor updates may seem futuristic and risk-prone, however when appropriately applied can be useful in remote medical sensor settings. While there are other legal and privacy issues involved with remote device updates, there are also many potential advantages. Some of the benefits using secure reliable updates in medical sensors include:

1. **No manual intervention** – Medical devices and sensors can remotely install, self-activate, reconfigure or even self-repair with new services and/or applications,
2. **Customization** – New clinical sensors can be added or dynamically configured and customized to the monitoring needs of the patient and health provider.
3. **Configuration** – Configuration provides the necessary control to a health provider to configure and control the operation of the different sensors in a body sensor network,
4. **Updates** – Updates enables applications to be dynamically updated in a sensor device or enable new features that improve the quality and reliability of the medical device.

Service flexibility. Service requirements in clinical monitoring imposes additional timeliness and priority constraints on the monitoring system. For example, some vital sign data have higher priority than others. MobiCare enables service components to dynamically self-activate, (re)-configure, update, and customize so as to suit the monitoring needs of the patient and the health providers. By using these mechanisms, the service components are able to effectively address the *time* and *priority* requirements in the monitoring of patients.

Wide-area link availability. The nature of health data available from a patient sensor network requires reliable, secure and time-bound data delivery to the provider servers. However, data

IEEE COMPUTER SOCIETY

delivery over wireless cellular links can be challenging. Such links are plagued by problems of high and variable round trip times (RTTs) and relatively low bandwidths. Links occasionally experience 'stalls' due to the loss of coverage (severe fades in the 'holes') and during handovers (device or patient mobility). Collectively, these issues exacerbate the challenges of reliable and time-bound data delivery over wireless cellular links. MobiCare overcomes these challenges through design of service protocols that helps to quickly adapt to the changing conditions of the underlying network. Additionally, MobiCare protocol design considers the 'nature' of the clinical data available and can effectively prioritize transmission of the health-critical data as required.

Data privacy and security. Data privacy and security can introduce myriad of problems in mobile healthcare. For example, patient health data can be misused by corporations (e.g., in deciding promotions), insurance companies (e.g., in refusing health coverage) etc.. Healthcare applications therefore must meet the stringent requirements of the Health Insurance Portability and Accountability Act (HIPAA) [17, 2] in the US and/or the data-protection ACT 1998 in Europe [16]. MobiCare addresses these issues with an infrastructure that enables: (1) secure data administration with healthcare providers, (2) sound network security, (3) secure sensing and monitoring devices, and, (4) stronger patient-provider authentication. Within these guidelines, MobiCare enables protection of personal health information while simultaneously enhancing the quality of patient care.

Description of Services

MobiCare defines mechanisms for health services and functions including service-specific parameters. We discuss some of these specific services next.

Protocol Definition MobiCare leverages HTTP to build its set of services. The main benefits are that it (1) allows reuse of the service infrastructure (servers), (ii) enables easy access to existing services and, more importantly, (iii) provides flexibility to compose new ones. Services in MobiCare are invoked using the standard HTTP protocol by submitting an HTTP request as a base URL (uniform resource locater) acting as a common access point for MobiCare services. The name of the service is then appended to the base URL as the final path component, and arguments to each service are encoded and appended as URL query parameters. Consider this example:

```
http://www.mobicare.net/services/Activation?Select
```

In this URL `Activation` corresponds to the name of the service and `Select` gives the service step for Activation. To interact with the servers a MobiCare client makes use of the standard **HTTP POST** method in the request header along with the URL meant for that service. The POST method allows data to be sent to the server in the client request itself. This is directed to a data handling program that server has access to (e.g., CGI, servlet). The data sent to the server is in the body of the request. After the server processes the POST request and headers, it passes the body to the server program specified by the URL.

Device activation service Device activation service in MobiCare enables client devices to self-activate and establish an account with the health provider. This is also known as remote registration (activation). Note that a device is typically activated once, however it may go through multiple registrations. Both activation and registration involve similar protocol steps.

Remote Configuration Service Configuration service allows flexible composition and control – new service parameters can be added or modified. Service-related parameters are stored in the persistent memory (flash) of the device.

Remote configuration service allows health providers to manipulate service-related parameters within the client device. For example, service-specific parameters manage and control settings of a body sensor network. Furthermore, it can help fix a problem in the client device due to misconfiguration of data and during remote dynamic device updates that require new settings.

Health data services Health data services are of two types: (i) download data services for health information downloads, and, (ii) upload data services for health diagnostic uploads. Download data service in MobiCare is used for health information downloads to the MobiCare clients. This service is particularly useful for chronically-ill patients that need regular feedback about their health and vital body signs. However, other than the health information, download data samples may also include software modules (e.g. updates or upgrades), new applications (e.g. micro-browser) or even content (e.g. video or jpeg images) for the device. On the other hand, health diagnostic upload services allow client devices to collect and upload patient health data to the health providers. The health data is collected by periodically monitoring the body sensors and uploading this data to the health providers. The data is analyzed by the health providers to provide any health feedback (using the download service) if required.

Dynamic code updates This is useful in three scenarios: (1) `Updates:` Enhancements that improve the quality or reliability of the device. This could be a release of an existing piece of system software for the device (e.g., a new MAC protocol for a new sensor) or a new medical application for the device, (2) `Upgrades:` Extensions that transform an existing client device into a new device to offer novel functionalities. (3) Applications: packages or even content downloaded by client devices. More discussion in the next section.

Elements of a MobiCare Client

MobiCare design is client-server based. In a typical setup, a MobiCare client will run in an embedded or wearable device such a wristwatch device [8], personal server [15] or even a cellular phone. The job of such a client device is to manage the body sensor network on instructions from the health provider, implement health services and provide for an external network interface with the cellular data network.

Server-end resources for health data infrastructure implements various services such as device activation, configuration, and health data services such as patient health diagnosis data uploads and health information downloads. Additionally, it implements

IEEE COMPUTER SOCIETY

tools to enable remote dynamic device code update services.

Dynamic Code Updates We believe that the chief novelty in MobiCare is a new technique that enables remote dynamic code updates applied to the native code of the client (clinical) device. The main advantage using dynamic code updates in clinical sensing devices is that it can accelerate deployment of new health-related features, services and applications in clinical devices in many different ways.

First, it permits easy sensor device customisation – new diverse clinical sensors can be added to the body sensor network without consequence for other sensors. These sensors will be dynamically configured with the body sensor network to the benefit of both patients and the providers. Second, it can provide the necessary control to a health provider to configure and operate sensors in a body sensor network setting. Finally, software updates enable new medical features to be incorporated in the device at run-time (e.g., a new sensor MAC protocol) or even help fix software bugs. Such updates or upgrades can improve the quality and reliability of the client device.

To support dynamic code updates, MobiCare adopts an approach similar to one used in the Microsoft's component object module (COM) model [13]. However, since most embedded real-time OS'es lack this functionality, we consider such solutions in an embedded client device. Note that support for dynamic code updates requires existing client modules to be able to offer functionality that can make it *independent* of other existing client modules. This additional functionality enables client modules to be able to *detach* their interface and end instantiation when needed. Moreover, modules in a client device are also able to release other residing modules' interface when so instructed.

MobiCare supports the following key features that enable dynamic code update functionality:

1. **Modular code –** modular code organization in clients
2. **Wrapper tool –** A compile and link time tool that can proxy-patch modules and prepare them for dynamic binding in client devices. Wrapping is the first step to prepare client modules for dynamic updates.
3. **Dynamic Loader (DynRTL) –** This feature enables two important functions in MobiCare: (i) run-time dynamic binding including dynamic loading and unloading of modules, and, (ii) dynamic updates/replacements of modules.
4. **PackBuild Tool –** A server-end tool that packages modules into a format intended for easy distribution and download by the MobiCare client devices.

These aforementioned features enable hot modular updates in MobiCare client devices. We briefly discuss these features.

Modules wrapping Wrapping is the first step to enable dynamic modular updates in MobiCare client devices. MobiCare offers a post-compilation tool called as the **wrapper tool** that externally compiles and links individual client modules and prepares them for dynamic binding. The chief function of this tool is to externally patch client modules with the "wrapper code" that can prepare them for modular code updates. In the first step it reads the module and intercepts unresolved references to external functions and and inserts the proxy functions that invokes dynamic binding with the DynRTL loader (discussed in the next section) to resolve these function calls. The second step of the tool patches function code that enables existing client modules to detach (by providing the *Detach* function) from the (retreating) module being replaced. The final step makes use of the *retreat* function that announces a module that it is being replaced, asks permissions for this replacement and performs the retreat actions.

Dynamic Loader (DynRTL) Dynamic loader (DynRTL) implements two key functions: (1) resolve inter module function references, and, (2) manage and maintain the client modules. DynRTL offers run-time support for resolving inter-module function references (binding) as well as functions for dynamic loading and unloading of modules. DynRTL module makes extensive use of the target OS system symbol table. The symbol table contains externally accessible functions in the system and its associated memory addresses. DynRTL is responsible for its relocation, registration of entry points for existing and any new modules with the system symbol table and if needed its removal as well.

Implementation and Status

We are currently building a proof-of-concept prototype for a MobiCare client. We are using an embedded real-time operating system and related development tools to prototype a MobiCare client device. We plan to show how we can provide such remote secure code updates services and other health-related services meant for a MobiCare client device.

These tests should help demonstrate the feasibility of the MobiCare architecture to enable new health services in a MobiCare client to be remotely installed, updated or reconfigured. In these tests we will use a wide-area cellular link and also potentially benchmark link availability, and then show how we can ensure reliable code updates for a clinical device. Finally, we will run appropriate web services to demonstrate the advantages of adopting an application layer approach.

Open Issues in MobiCare

In this section we discuss some of the important open issues relevant in the context of MobiCare.

Security Issues A continuous concern in a health infrastructure like MobiCare is that of security. In fact, several open questions remain. How safe and dependable are these clinical devices that are worn or implanted? How do we ensure confidentiality in a body sensor network? For instance, an ECG signal of a patient is jammed, error'ed or modified such that wrong diagnosis and treatment are prescribed which may cause even death. Such issues are also discussed in detail in [14].

Note that it is difficult to address all the open security issues in a system like MobiCare. We do however review some of the key ones. For example, data integrity across multiple clinical sensors can be ensured using shared keys in such client (clinical) devices.

MobiCare client and server security can be addressed at the application-layer itself. For example, Wireless Application

IEEE COMPUTER SOCIETY

Protocol (WAP) based Wireless Transport Layer Security (WTLS) protocol could be used to provide privacy, data integrity, and authentication over the cellular link [7]. WTLS also closely resembles Secure Socket Layer (SSL) protocol, yet is optimized for use over cellular links and also suits resource-constrained mobile devices. Further, patient health information may introduce additional security vulnerability. Any potential leakage of the health data information may be prevented by ensuring that only clients from select cellular networks and provider-authenticated personnels connect to the provider servers.

Legal and Privacy Concerns A remote patient health monitoring system like MobiCare may impose additional restrictions. For example, it may cause unwanted intrusion in the lives and privacy of patients. Although prior studies in this area demonstrated that this is not the case for most patients [11], our ongoing research work investigates such issues further. MobiCare also introduces new legal and security-related challenges [17]. For example, in some developed countries health providers may not get prior approval for dynamic software updates before they could be used in clinical trials. However we believe that the service flexibility offered in MobiCare to chronically-ill patients far outweighs the legal issues and hurdles involved in its deployment, once health services and applications are test-trialled by a provider on a medium to large-scale basis. Moreover, the flexibility of deploying new health-related services and application in MobiCare is its primary appeal, and new guidelines could be enforced for secure and reliable dynamic updates and deployment of security solutions that offer acceptable trade-offs between flexibility and security.

Some Related Work

Early clinical trials to gain insights into how medical systems may help patients evolve in mobile settings were conducted in the mid-nineties by the National Institute of Health (NIH) in the Mobile Telemedicine project [11]. Key lessons learned from these tests were: (1) reliable, high-bandwidth wireless data communications is difficult, (2) transmission of critical patient data during emergencies can make significant difference in patient outcomes, and, (3) remote patient diagnosis is difficult. However these testtrails involved second generation GSM cellular networks, and the various limitations seen in the Mobile Telemedicine project have been overcome (to a large extent) by the recent advances made in the areas of clinical sensors, wearable computing and mobile cellular communications.

Dedicated clinical sensors have long been used (though rather restrictively) in various medical settings. Such sensors are examples of *small-form* wearable devices that accomplish certain dedicated tasks. However, sophisticated wearable device such as the IBM's Linux Wrist Watch [12, 8, 9] go way beyond the potential of the such wearable clinical sensors. Similarly, amon [16] is a wearable wristwatch-style medical monitoring and alert system for patients. We believe that such wearable devices combine the necessary functionalities to accomplish the different tasks in medical care settings. Therefore, barring few simple modifications to such wristwatch devices, porting MobiCare client functionality should be straightforward. For patients a wearable device like the wristwatch not only monitors 'health-critical' data, but can also collect, store and perform periodic uploads with the health servers.

MobiCare shares many of the goals and objectives with the Patient Centric Network (PCN) project at MIT [6]. This project is developing a prototype to address service-specific issues for patient sensor network. The system will consists of software components running on general purpose computers and networks to link users with a variety of medical sensors and actuators. The goal in PCN is to accelerate innovation, decrease cost, and improve the clinical quality of medical care.

Conclusions

The vast opportunity in the 'point-of-care' access and the capture and transmission of patient information will continue to drive the healthcare industry towards increased mobility. The importance is in the shifting awareness that mobility in healthcare settings increasingly refers to – the mobility of sensor/actuator devices, the healthcare providers (health 'outsourcing') and of the patient (users) themselves. MobiCare leverages the point-of-care patient access to offer important benefits like quality healthcare, a programmable service architecture, flexible service composition and a full-scale medical systems integration.

MobiCare is an ongoing project and much work remains to be done. Besides a proof-of-concept prototype, we are also in the process of investigating other long-term, challenging research problems in MobiCare including the body sensor network security, reliable and secure sensor code updates and upgrades, the potential legal hurdles involved and the privacy issues that arise with dynamic remote code updates. We will address all of these above challenges to enable a ubiquitous computing and communications infrastructure for large-scale, pervasive healthcare services.

References

[1] Health Frontier Inc. http://www.healthfrontier.com.
[2] HIPAA. http://www.hipaa.org.
[3] Linde Medical Sensors AG. http://www.linde−ms.ch.
[4] Nonin Medical Inc.. http://www.nonin.org.
[5] Numed Holdings Ltd.. http://www.numed.co.uk.
[6] Patient-Centric Network at MIT/LCS. http://nms.lcs.mit.edu/projects/pcn/.
[7] WAP Forum. http://www.wapforum.org.
[8] C. Narayanaswami, et al. IBM's Linux Watch: The Challenge of Miniaturization. *IEEE Computer*, 2002.
[9] C. Narayanaswami, M. Raghunath, N. Kamijoh, T. Inoue. What would you do with a Hundred MIPS on Your Wrist? In *IBM Research Report (RC 22057)*, 2001.
[10] D. Malan, T. Fulford-Jones, M. Welsh, S. Moulton. CodeBlue: An Ad hoc Sensor Network Infrastructure for Emergency Medical Care. In *Proc. of International Workshop on Wearable and Implantable Body Sensor Networks*, 2004.
[11] Final Report – Mobile Telemedicine Testbed For National Information Infrastructure. National Institute of Health, Aug 1998. Project N0-1-LM-6-3541. http://collab.nlm.nih.gov/tutorialspublicationsandmaterials/telesymposiumcd/bdmfinal.pdf. 1997.
[12] IBM Linux Wrist Watch. http://www.research.ibm.com/WearableComputing/.
[13] Microsoft COM. http://www.microsoft.com/com/.
[14] R. Anderson. System Security for Cyborgs. In *International Workshop on Wearable and Implantable Body Sensor Networks*, 2005.
[15] R. Want, T. Pering, G. Danneels, M. Kumar, M. Sundar and J. Light. The personal server: Changing the way we think about ubiquitous computing. In *Proc. of the 4th international conference on Ubiquitous Computing (UBICOMP)*, 2002.
[16] U. Anliker, J. A. Ward, P. Lukowitcz, et. al. AMON: A Wearable Multiparameter Medical Monitoring and Alert System. *IEEE Transactions on Information Technology in Biomedicine*, 8(4), 2004.
[17] V. Stanford. Pervasive Health Care Applications Face Tough Security Challenge. *IEEE Pervasive Computing*, 2002.