IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

A Framework for Multimedia Medical Data Integration of Adaptive Mobile Object in Heterogeneous System

Roselin.T[1], Priya.S.V[2]

[1]M.E. II Year, Department of Computer Science and Engineering, Sriram Engineering College, Perumalpattu – 602 024

[2]Associate Professor and HOD, Department of Computer Science and Engineering, Sriram Engineering College, Perumalpattu – 602 024

Abstract

With the advent of 4G and other long-term evolution (LTE) wireless networks, the traditional boundaries of patient record propagation are diminishing as networking technologies extend the reach of hospital infrastructure and provide on-demand mobile access to medical multimedia data. However, due to legacy and proprietary software, storage and decommissioning costs, and the price of centralization and redevelopment, it remains complex, expensive, and often unfeasible for hospitals to deploy their infrastructure for online and mobile use. This paper proposes the Spark Med data integration framework for mobile healthcare (m-Health), which significantly benefits from the enhanced network capabilities of LTE wireless technologies, by enabling a wide range of heterogeneous medical software and database systems (such as the picture archiving and communication systems, hospital information system, and reporting systems) to be dynamically integrated into a cloud-like peer-to-peer multimedia data store. Our framework allows medical data applications to share data with mobile hosts over a wireless network (such as WiFi and 3G), by binding to existing software systems and deploying them as m-Health applications. Spark Med integrates techniques from multimedia streaming, rich Internet applications (RIA), and remote procedure call (RPC) frameworks to construct a Self-managing, Pervasive Automated network for Medical Enterprise Data (Spark Med).

Index Terms**—***Automated systems, biomedical engineering, handheld computing, m-Health, middleware, mobile communication, notebook computers, telemedicine***.**

I. INTRODUCTION

MOBILE healthcare, or m-Health, is one of the fastest growing areas of healthcare computing [1]. As electronic health records become commonplace, and the rapid uptake of mobile and handheld devices puts powerful portable computing devices into the hands of an ever-increasing proportion of the populace (inclusive of those in low-income and disadvantaged areas, the stage is set for future wireless communication technologies to revolutionize patient care by making health ser-vices both portable and interoperable.

The next generation of networking is here, in the shape of new 4G and long-term evolution (LTE) wireless technologies such as WiMAX, which are all IP-based heterogeneous networks aimed at vastly expanding the accessibility and usability of any internet-connected system. LTE technologies are portable, lightweight and nonproprietary, and provide mobile devices with access to integrated communications standards that have low transmission costs and rich multimedia support. A major goal of LTE wireless technologies is the provision of personalized, reliable wireless data services that can allow even simple hand-held devices to easily make use of multiple multimedia data streams at the same time.

Unfortunately, taking advantage of the benefits of LTE to deploy mobile healthcare technology entails major costs (in money, time, and computing resources) for any hospital infrastructure, and consequent problems such as resistance to change and the difficulty in using new systems. The most significant problems, however, result from: 1) the limited scope of access to data in proprietary hospital infrastructure systems; 2) the need to replace or decommission medical applications and data services if they do not support a networked healthcare model; 3) the storage and post processing requirements that keep medical data from becoming portable; and 4) the lack of a centralized repository or common standard for most healthcare data.

The motivation of this paper has been to address the afore-mentioned problems, to allow medical data applications to share data with mobile hosts over a wireless network by binding them to existing software packages and allowing them to operate (or be remotely operated) as m-Health applications. SparkMed integrates techniques from mobile technologies such as multi-media streaming, rich Internet applications (RIA), and remote procedure call (RPC) frameworks to construct a Self-managing, Pervasive Automated networRK for Medical Enterprise Data (SparkMed).

Our SparkMed design ensures: 1) minor interference with the regular operation of the host medical software it is bound to (given the importance of its continued functioning); 2) min-imal overhead to make sure that the host system's performance remains unaffected while still allowing it to be utilized interactively across any IP-based connection; as well as 3) providing core functionality to limit the cost and scope of reprogramming.

SparkMed provides a number of automated, self-configuring services including discovery, data monitoring and synchronization, thread pooling for remote functions, collaborative remote control capability, and transcoding to web-based standards. We demonstrate this functionality with a prototype SparkMed sys-tem that implements these services for a simulated nuclear medicine workflow environment modeled upon the clinical information system in the Department of PET and Nuclear Medicine at our partner hospital, the Royal Prince Alfred Hospital (Camperdown, N.S.W., Australia). Performance was measured when propagating multimedia medical data to mobile hosts over three distinct network configurations: remote access over WiFi, roaming access via 3G, and "headless" use with an all-mobile network. We observe the performance of the mobile-based medical software, and measure the amount of overhead imposed upon the medical multimedia software we use as our data source.

II. BACKGROUND

Many successful healthcare applications based on the picture archiving and communication systems (PACS), and us-ing ubiquitous computing technologies, have been presented in the literature. For instance, projects such as Web-PACS and PACS flow and software packages such as the diagnostic image viewer OsiriX allow for inter-institutional and web-based access to multimedia medical data. However, numerous heterogeneous database and diagnostic systems supplement PACS, and they typically require greater computing resources than those that are available on mobile devices There have been many attempts to develop telemedicine solutions that take advantage of mobile devices; however, the majority rely greatly on dedicated infrastructure, which limits their functionality and the scope of their deployment. There is already speculation that the next generation of PACS is moving toward a cloud-computing-based system, with its data distributed within a net-work of secure online repositories. Known in the literature as hosted PACS, offsite PACS, or PACS Software as a Ser-vice (SaaS), a number of commercial and research systems are already making major steps in this direction . With increasing retention requirements, blurring distinctions be-tween data storage and archiving, and an ever-increasing volume of medical data, the cost-effectiveness of cloud-based PACS and its inherent benefits for accessibility and disaster recovery make such systems very attractive to hospital administrators.

A. Service-Oriented Architectures

Service-oriented architectures (SOA),are a common answer to this problem: an approach which consists essentially of breaking down existing software systems into inter-operable web services, and leveraging these to create complex medical applications and deploy them over a cloud network. Such an approach combines the benefits of both an in-house and a cloud solution. As a design principle, SOA provides an ideal solution for a healthcare environment, which contains nu-merous heterogeneous data services and isolated, specialized workstations that are difficult to interface with one another. Further, there is evidence that SOA can provide the necessary scalability and sustainability to support a healthcare information system .

In practical terms, however, such SOA implementations rely on the availability of componentized, interoperable web services that support a common syntax and semantics. XML, SOAP, MPEG-7, and MPEG-21 are all powerful standards for semantically rich delivery of data and metadata between web services in a service-oriented architecture. However, these services must be manually developed in a healthcare context; there is no common, accepted middleware standard for doing so; an in-house SOA entails significant overhead; and there is substantial expense incurred in integrating a pre-existing infrastructure to function as an SOA. As such, it is a substantial organizational challenge for any healthcare provider to adopt such an approach. Further, since it is primarily a development methodology to be followed when initially creating a system, a service-oriented architecture can be unfeasible to create from an existing system, largely due to custom-built, proprietary, and legacy components.

B. Prevalence of Mobile Devices in Healthcare

Most specialist healthcare providers own at least one mobile device, such as a SmartPhone, and that number is rising These mobile devices bring together all of the benefits of pagers, PDAs, and mobile phones, but the social networking and media functionality that is becoming ubiquitous on these devices allows health professionals to expand their practice and their relationships with patients Further, these devices are showing promise as emergency teleconsultation devices, making true telepresence for medical practitioners possible without even the need for specialized hardware. The current popularity surge of tablet devices such as the Apple iPad, and their increasing popularity in hospitals, has made form-factor considerations for handheld and mobile devices less important. In light of these facts, there is no longer as much of a bias against mobile devices in the healthcare sector as there was in the past.

C. Integration Methods

Numerous tools and technologies, however, exist for integrating such components to function as interoperable, query-able services compatible with mobile and web-based clients. These have in many cases already been applied to electronic patient records (EPRs) and medical enterprise data. Such techniques include the use of RPC frameworks such as

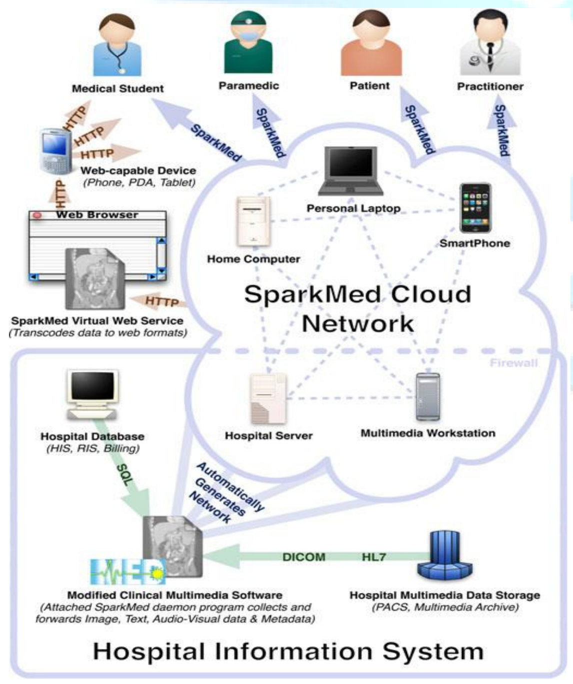
Medical Student
Paramedic
Patient
Practitioner
HTTP
HTTP
HTTP
SparkMed
SparkMed
SparkMed
SparkMed
Web-capable Device
(Phone, PDA, Tablet)
HTTP
Web Browser
Personal Laptop
SmartPhone
Home Computer
HTTP
SparkMed Virtual Web Service
(Transcodes data to web formats)
SparkMed Cloud
Network
Firewall
Hospital Database
(HIS, RIS, Billing)
Hospital Server
Multimedia Workstation
SQL
Automatically Generates Network
DICOM
HL7
Modified Clinical Multimedia Software
(Attached SparkMed daemon program collects and forwards Image, Text, Audio-Visual data & Metadata)
Hospital Multimedia Data Storage
(PACS, Multimedia Archive)
Hospital Information System

Fig. 1. SparkMed cloud network, as generated automatically by our frame-work. Medical data are forwarded to mobile clients by means of a self-configuring, multicomputer network.

CORBA federated database systems, software agents, semantic data sources such as ontologies, and various forms of wrap-per generation and encoding designed to translate dynamically between medical formats.

D. Our Related Work

Our previous research into mobile and web-based medi-cal imaging technology was developed to address the need for mobile healthcare solutions by enhancing mo-bile telemedicine. Our new SparkMed framework integrates and extends our group’s previous work developing mobile health-care solutions, such as the mobile, active medical protocol (MAMP) and our mobile “INVOLVE” dual-modality di-agnostic workstation.

III. METHODOLOGY

In this section, we will go over the individual parts of our SparkMed framework for dynamic integration of multimedia medical data. Fig. 1 illustrates the conceptual outline of our net-work. This network is composed of devices inside and outside of hospitals and medical institutions, both desktop and mobile, and a series of web servers that can be either Intranet based or Internet based. Fig. 2 illustrates the hierarchy of networking layers used by our prototype. This hierarchical architecture is similar to some existing proposed middleware systems for remote application control and medical data propagation, but we introduce several improvements. First, our network architecture is automatically self-generated without the need for user input or even network support in the host application. Second, we make use of a novel daemon technique to run a transparent, attached process and adapt network input and output to mimic normal usage, thus allowing compatibility with even legacy medical software. A similar technique is employed by systems that implement autonomic computing techniques to retrofit legacy software with new functionality. However, SparkMed only implements a subset of the official definition of autonomic computing as necessary to ensure that it can make effective data-synchronizing decisions and react intelligently to changes under network conditions.

The following section describes the software architecture and design of SparkMed, as well as the networking techniques it employs, in more detail.

A. Overall Architecture Design

Individual mobile- and desktop-based client nodes are the main components of our SparkMed architecture. A SparkMed client node is a stand-alone daemon program running inside a piece of medical software on any Internet-capable computing device. The attached medical software may in turn be itself running inside a web browser.

The individual SparkMed daemons use a connectionless Zeroconf service discovery approach to connect to other daemons. All SparkMed daemons are identical, but their host applications determine which data items they share or provide, and this information is forwarded whenever the daemons connect to one another Individual nodes retain memory of other available nodes, keep an index of what data each provides or accepts, and are able to reconfigure their network to recover from service interruptions with a minimal disruption in functionality.

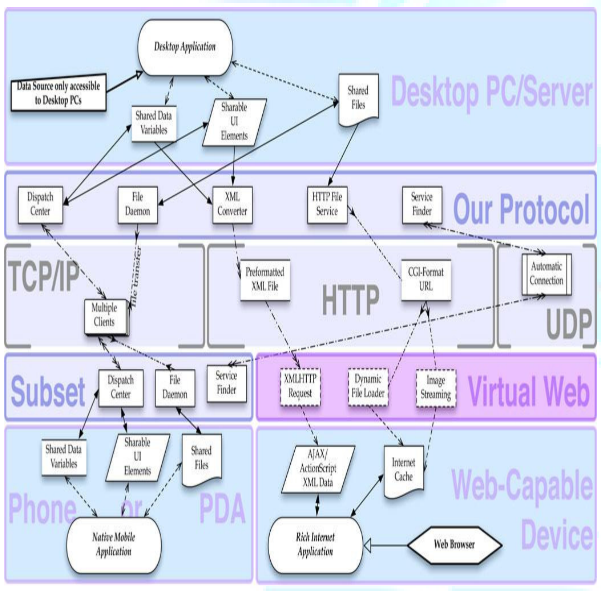

Fig. 2. Networking architecture of our prototype, showing the relationship between data at the hospital (top), and the means by which it is conveyed to mobile and browser-based hosts (bottom).

B. Cloud Networking and Centrality Measurement

Having detected one another, SparkMed nodes communicate to generate a medical data "cloud": effectively a query-able centralized repository integrating every data item in the SparkMed network. The centrality of each node is calculated numerically in order to choose a central "server" node, with the key criteria for determining the centrality of a node being based on reachability, network access time, and security/storage capability. This calculation is performed by first filtering out all nodes which do not meet the data storage or security requirements of the active data sources, and those on peripheral or unreliable physical network connections (as determined over time using observed reachability from each node, and frequency of disconnection). If G is the graph representing SparkMed's network of interconnected nodes, then for any given node the *delta centrality*, *C* , is calculated using

$$C = \frac{(\Delta P)_i}{P} = \frac{P[G] - P[G^-]}{P[G]} \qquad (1)$$

where *P* is an observed ping value (propagation time of a simple data packet with and without the given node being used to for-ward the request) and $(\Delta P)_i$ is the *i*th variation of *P* where this node is no longer used to forward data as part of the SparkMed network. By G^- we indicate a variation upon the SparkMed net-work graph G, obtained by removing this node (and hence the graph edges it represents). This represents a similar approach to, using observed ping values as a measure of network cohesiveness, to specify the importance of each node to the overall network. This implies that the central node of a SparkMed net-work will change over time, although in practice, recalculation only occurs when warranted by failure or a loss of connectivity.

This measure determines which node is considered central. Data synchronization is achieved by means of a priority-based synchronization system that propagates any data changes to this central node, which in turn updates the entire network of nodes. Each non-central node registers data of interest, representing medical data types the node consumes. The network then ensures that each node is kept synchronized with its respective data source(s) at all times. Conflict resolution is applied in the case of incompatible changes, weighted by the access credentials of the responsible user (where applicable) and the time the change was made. Timing is calculated via POSIX timestamps to an accuracy of approximately 5 ms, while resynchronizing device time at regular intervals to account for clock drift.

Whereas SparkMed is a single integrated service (i.e., there is only one SparkMed network), each application that contains the SparkMed daemon may consume and synchronize different data, and as such numerous separate services can be operating within the SparkMed cloud at any given time. Although each discrete set of application data can be considered a separate web service, there are overlaps between the data used by different applications (such as between mobile and desktop versions of the same application) and a single application may use data from multiple sources (e.g., PACS imaging and hospital information system (HIS) patient data). As such, the number of central nodes is dependent on the number of discrete data sources, which may form a part of any number of applications or services. Specifically, if no single central node is capable of supporting, accessing, or storing all of the data types required, multiple central nodes will ensue, each of which is responsible for a different category of data.

C. Implementation Languages

The majority of autonomous threads operating within the network architecture, including daemons and the RIA interface, were developed using the Java programming language. This choice was made due to the write-once-run-anywhere promise of the Java virtual machine, and for compatibility with CORBA and potential integrability with other leading pervasive medicine frameworks such as the Java Context Awareness Framework. For maximum compatibility with mobile and web-based devices, subsets of the SparkMed daemon were also implemented on the Apple iOS and Adobe Flash platforms. iOS was chosen because it is necessary to run natively on some of the most advanced and prevalent handhelds in the healthcare market. As to non-iOS devices, although Java may be considered ubiquitous on most other handhelds, our investigation of the literature [36] led us to choose the Adobe Flash IDE as the implementation platform for our RIA interface. SparkMed data components have been implemented for the Java, Objective-C, and several RIA environments.

D. Shared Data Propagation Functionality

The lightweight SparkMed daemon interacts with SparkMed data components placed inside the host medical software (referred to herein as “Shared Data items”). These are simple data storage and monitoring objects which can replace (or extend) the standard primitive types and user interface (UI) components of the underlying development framework. These components’ functionality is equivalent to the corresponding primitive types or UI components, except that they are thread-safe, will automatically synchronize with networked equivalents, and provide a notification feature so that the base software can watch for changes in value. In the SparkMed UI components, data changes due to synchronization are treated as user input (allowing the base application to respond as normal). This ensures that the functionality of SparkMed does not interfere with the operation of the base medical software. The SparkMed daemon itself can be compiled directly into the application as a library, and need only be invoked in the code. Once started, its operations are automatic.

Where a suitable SparkMed data component is not available, the daemon can instead interface with the software indirectly. It is currently capable of SQL-requests and provides bridging functionality for a variety of medical database and PACS software, including Filemaker Pro. In cases where this is necessary, the database is treated as a subnode with no daemon of its own, and the SparkMed daemon active on the same machine becomes responsible for the synchronization of its data as normal.

E. Overlay Network and Communication Standards

Our protocol’s network architecture is analogous to the concept of a “personal overlay network,” as discussed in [37], in that it straddles the existing network topology separating hospital systems and handheld devices, using these devices as nodes in a peer-to-peer network. The loss of any one SparkMed node should not jeopardize the user’s data or the functionality of the system, as the network as a whole should be able to adapt to its removal (by sourcing data elsewhere, or attempting reconnection).

SparkMed nodes use a variety of medical and internet standards in their operations, as appropriate. The daemon itself reads medical information directly from the host application, and hence supports DICOM and HL7 formats and protocols as long as the base application does so, as well as containing innate support for consumer image and video formats. Mobile and desktop computer nodes communicate with one another via our own custom communication protocol over TCP/IP. Images are transmitted in JPEG format.

G. Operational Details

Fig. 2 depicts the layers of the SparkMed protocol. The top represents a standard medical software application, running on a consumer PC or server. Irrespective of its own data processes, the data it loads are stored in memory as program variables and files, and the user manipulates the application via the standard UI components of the operating system. Our framework substitutes these variables, UI elements, and files with equivalent Shared. Data accessible to the SparkMed daemon. The daemon implements its own suite of services and uses its own protocol on top of the base application as shown in the figure. Each daemon connects to other instances of itself, by using the connectionless UDP protocol to probe the network for other SparkMed nodes.

When such nodes are found, they may be one of three types: another piece of medical software implementing SparkMed, or a subset of SparkMed running as a mobile application or RIA. In the case of other desktop applications or mobile native applications, SparkMed establishes a TCP/IP connection and synchronizes the UI elements, variables, and file data between nodes. In the case of web applications, SparkMed implements an AJAX-like interface for XML-based communication of data, and a simple HTTP server for serving files, so as to allow for easy web-based access to the same data. Addressing of data is more complex for web-based applications, since the data are not directly synchronized. SparkMed implements a simple URI model to allow these applications to simply treat remote data as if it were a static file (or CGI script) on a web server, and reference it via dynamically assigned URIs.

Each daemon waits for its host application to start nor-mally, before creating an index of shared data used or created by this application. Once all data have been registered, the daemon seeks out other SparkMed nodes on the network, compares this application's data index with theirs, and hence creates an access list of compatible nodes. These nodes are registered inside the daemon's internal Dispatch Center as watchers for the relevant shared data, and state information is exchanged so as to synchronize these data with the remote daemons. Thereafter, these watchers are notified whenever the data change, and watched for relevant status updates.

IV. CONCLUSION

This paper presented SparkMed, a framework to enable mo-bile access to multimedia medical data to a wide range of Internet-capable and mobile devices. We have outlined the functionality of the system, demonstrated its capability to inter-actively deploy medical multimedia systems to mobile device clients, and intelligently synchronize and propagate medical data from a variety of heterogeneous sources in a convenient, reliable manner without appending significant overhead to the underlying process.

Our prototype and case scenario evaluated the effectiveness of the SparkMed architecture in an environment designed to simulate a real hospital and telemedicine setting. Within the context of our simulated radiological workstation, our prototype demonstrated highly interactive usability and low overhead cost requirements, proving its suitability and effectiveness in similar hospital contexts.

REFERENCES

[1] Foundation for the National Institutes of Health. (2009). *The Inaugural mHealth (mobile health) Summit*, Washington D.C. [Online]. Available: http://www.fic.nih.gov/news/events/mhealthsummit.htm

[2] D. Vatsalan, S. Arunatileka, K. Chapman, G. Senaviratne, S. Sudahar, D. Wijetileka, and Y. Wickramasinghe, "Mobile technologies for enhancing eHealth solutions in developing countries," in *Proc. 2nd Int. Conf. eHealth, Telemed., Soc. Med., eTELEMED 2010, Includes MLMB 2010; BUSMMed 2010*, pp. 84–89.

[3] R. Istepanian, C. S. Pattichis, and S. Laxminarayan, "Ubiquitous m-health systems and the convergence towards 4G mobile technologies," in *M-Health: Emerging Mobile Health Systems*. New York: Springer-Verlag, 2005, pp. 3–14.

[4] J. A. Hernandez, C. J. Acuna, M. V. de Castro, E. Marcos, M. Lopez, and
N.Malpica, "Web-PACS for multicenter clinical trials," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 1, pp. 87–93, Jan. 2007.

[5] I. Balasingham, H. Ihlen, W. Leister, P. Roe, and E. Samset, "Communi-cation of medical images, text, and messages in inter-enterprise systems:
A case study in Norway," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 1, pp. 7–13, Jan. 2007.

[6] A. Rosset, L. Spadola, and O. Ratib, "OsiriX: an open-source software for navigating in multidimensional DICOM images," *J. Digital Imag.*, vol. 17, no. 3, pp. 205–216, Sep. 2004.

[7] A. Kailas, C. Chong, and F. Watanabe, "From mobile phones to personal wellness dashboards," *IEEE Pulse*, vol. 1, no. 1, pp. 57–63, Jul.–Aug. 2010.

[8] 2007 IBM Report on Health Care. (2007), "Healthcare 2015: Win-win or lose-lose? A portrait and a path to successful transformation," IBM Insti-tute for Business Value, pp. 1–8. [Online]. Available: http://www-935.ibm.com/services/us/gbs/bus/pdf/healthcare2015-win-win_or_lose-lose.pdf

[9] J. Philbin, F. Prior, and P. Nagy, "Will the next generation of PACS be sitting on a cloud?" *J. Digital Imag.*, vol. 24, no. 2, pp. 179–183, Apr. 2011.

[10] C. Costa, C. Ferreira, L. Bastiao, L. Ribeiro, A. Silva, and J. L. Oliveira, "Dicoogle—An open source peer-to-peer PACS," *J. Digital Imag.*, vol. 24, no. 5, pp. 848–856, Oct. 2010.