

Original Research

Design and Implementation of a Message-Based Regional Telemedicine System to Achieve High Availability and Scalability

Xiaoyang Ren, MD,¹ Zhenbo Wang, MD,² Yajie Wu, MD,¹ Yushen Li, MD,¹ Mingxia Chen, MD,¹ Yunkai Zhai, PhD,^{2,*} and Yuhong Li, MD^{1,*}

Departments of ¹Information and ²Telemedicine, The First Affiliated Hospital of Zhengzhou University, Zhengzhou, People's Republic of China.

Abstract

Background: Different levels of telemedicine systems have been built across China. To share high-quality medical resources and conduct centralized management of telemedicine systems, a regional telemedicine system (RTS) (such as a provincial level system) has been developed to integrate with small-scale telemedicine systems. Although the established systems offer integration services, they are tightly coupled systems, and not easily integrated with new systems. Meanwhile, with the increasing of input/output, it is difficult for them to run with high scalability, considering the cost of architecture redesign and further development. This article presents the design and implementation of regional integration system through a study in Henan, China, mainly aimed to integrate with heterogeneous small-scale telemedicine systems and provide high efficiency.

Methods: A provincial telemedicine system and some city-level telemedicine systems have already been established. The provincial system has been built to act as a regional integration system to connect city-level systems. Adopting message-based technology, the provincial system achieves high availability and high scalability, respectively, through LevelDB + ZooKeeper and multicast.

Results: The system achieved the centralized management of established telemedicine systems without restructuring their framework, improving high availability of RTS when one ActiveMQ service node in a group failed, and it did not negatively influence normal business logic when adding a new service node. At the same time, two "Master" state ActiveMQ service nodes provided services simultaneously, which enable the RTS to achieve high scalability.

Conclusions: The message-based regional integration system enriched the RTS with high availability, easy extensibility,

and provided a convenient way to integrate new small-scale telemedicine systems.

Keywords: telemedicine, technology, e-health, information management

Introduction

As information and communication technologies have advanced forward for decades, it is an unprecedented opportunity to overcome traditional restriction that the medical services provider and the recipient must be present in the same place and at the same time. Healthcare aids have been improved under this trend. Information technology (IT), including telemedicine, is a key to improve the regional health service.^{1,2} Telemedicine, which involved with medicine, information, and telecommunication technologies, is likely to be an important part of the revolution that could have the greatest impact on healthcare delivery. Telemedicine is an integration of diverse technologies and clinical applications, and is characterized by the use of communication technologies and IT to provide medical services to individuals who are far away from the medical services provider.^{3,4}

Telemedicine systems have been booming for several decades, and most telemedicine systems have been developing in industrialized countries, especially the United States, Australia, and United Kingdom.⁵ Besides, Finland, Russia, and Latin America also have launched telemedicine programs.^{6,7} China's telemedicine activities began in mid-1980s. Telegraph and e-mail are mostly selected as transmission media of telemedicine systems. In recent years, telemedicine in China has developed quickly with the rapid growth of telecommunication network,⁸ and provided services such as teleconsultation, remote education, and videoconference. Most recently, to promote the advancement of telemedicine, regional collaborative medical service system (RCMS) has been constructed.⁹ Through RCMS, the efficiency of telemedicine activities has been promoted among beneficiaries, such as hospitals, urban and rural areas, doctors, and common people.¹⁰

*Both authors contributed equally to this work and should be considered co-corresponding authors.

RCMS maximizes the use of medical resources, rationalizes the allocation of resources, and lower medical costs.¹¹ However, it costs much to build RCMS; furthermore, it is difficult to achieve scalability while constructing a new lower level telemedicine system.

The aim of the present article is to propose a message-based regional telemedicine system (RTS). When building multilayer RTS, it provides high availability. The message-based RTS extends easily when RTS integrates with a lower level telemedicine system.

Methods

SCENARIO ANALYSIS

One RTS application is reflected in the synchronization of personal information about medical experts, such as their location and specialties. Through RTS, users of other small-scale telemedicine systems belonging to the same RTS can apply information to consult the expert or experts they prefer. Scenario analysis of RTS lists as follows:

RTS connects small-scale telemedicine systems in the region. Each area only defers to local situation to develop medical information. Some areas have developed telemedicine systems and already joined RTS, while some areas have their own systems and will join RTS, and even some areas have no telemedicine system (Fig. 1).

Synchronization of personal information about experts. Local experts of the joined small-scale telemedicine system have registered with their personal information in RTS (uplink synchronization), and then RTS shared information with other small-scale systems in the same RTS (downlink synchronization).

Cross-area teleconsultation requesting. Taking teleconsultation as an example, an applicant could read the information about registered experts, request appropriate candidates for teleconsultation, and then send the required information of patient. The report will be sending back to the applicant once experts perform diagnosis.

Small-scale telemedicine system joins RTS. The system uploads a number of personal information about local experts to RTS, and then RTS executes a batching process to store and share information with other systems in the RTS.

RTS supports multilanguages. Small-scale telemedicine systems may come from different suppliers at different time; moreover, they are built mostly by different languages. Therefore, RTS should provide supporting in a variety of programming languages, or use a general protocol that supplies multilanguages. If so, RTS could ensure that these systems can normally operate without changing programming language, and there is no need for these systems to adjust existing processes.

GENERAL DESIGN

One of main purpose of providing message-based service is to avoid restructuring existing structure of small-scale telemedicine system (Fig. 2).

Avoiding data congestion is another merit of the adoption of message-based service. As mentioned in the Scenario Analysis section, when a small-scale telemedicine system joins RTS, a full synchronization is required. Personal information of all available local experts is registered in the joined system and shared in RTS to achieve uplink synchronization. In the same way, RTS shares personal information about experts with the system to achieve downlink synchronization. In this case, what is going to happen is the accumulation of large volumes of data. The reason lies in the high performance of data persistence of ActiveMQ that is

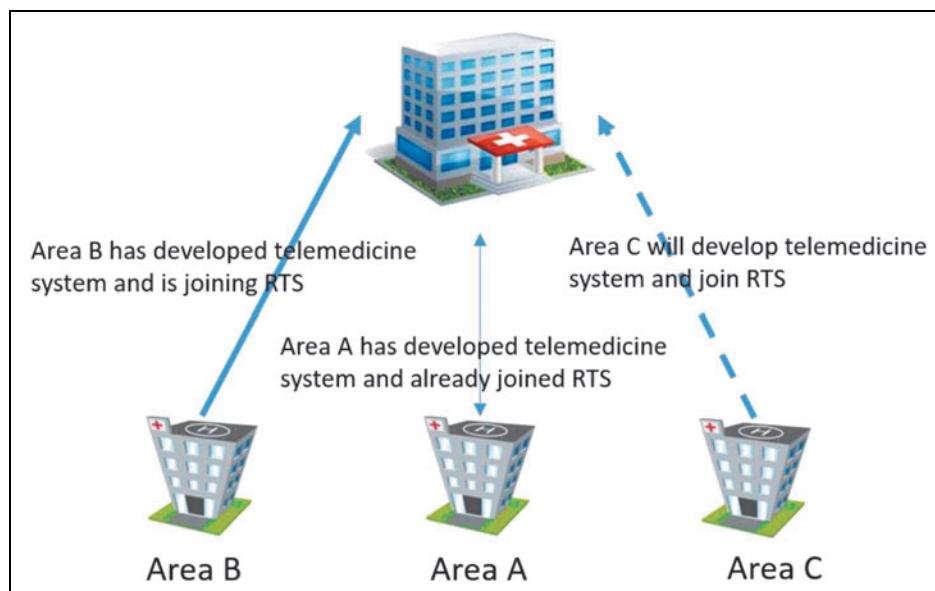


Fig. 1. Area A's telemedicine system has joined RTS; Area B's telemedicine system is joining RTS, such as debugging interface modules and synchronizing information; Area C will develop its own telemedicine system. RTS, regional telemedicine system.

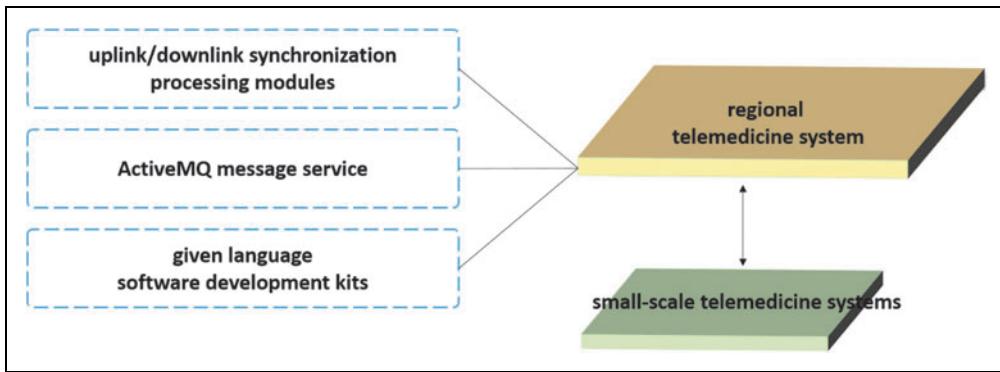


Fig. 2. General design diagram of RTS. RTS provides some given language software development kits, facilitating small-scale telemedicine systems joining RTS. Small-scale telemedicine system calls the appropriate development kit, without changing its internal architecture and programming language. The provided kit offers the following main functional areas: it encapsulates behaviors that are triggered by connecting message-based service, normalizes the uplink and downlink message format, encapsulates encryption and decryption protocols of message, and encapsulates the procedure of sending and subscribing message. RTS adopts message-based service to avoid data congestion. It benefits from high performance in data persistence of ActiveMQ that we choose as message-based service middleware, the stress of data synchronization will not pass on to upper system. RTS has uplink and downlink synchronization processing modules, based on ActiveMQ message service, to process messages from small-scale telemedicine systems, which use software development kits, provided by RTS.

chosen as the message-based service medium. The stress of data synchronization also will not be passed on to upper system.

Based on the scenario analysis of small-scale telemedicine system that joins RTS, the biggest problems are as follows: they have different system architecture, programming language, and development team. To ensure the safety and performance of the joining process, RTS provides some given language software development kits (Java and C#) and relevant documents.

The provided software development kit offers the following main functional areas: it encapsulates behaviors, which are triggered by connecting message-based service, normalizes the uplink and downlink message formats, encapsulates encryption and decryption protocol of message, and encapsulates the procedure of sending and subscribing messages.

Finally, to guarantee that all uplink and downlink messages can be handled by the target system, all messages should be in PERSISTENT state.¹² Besides, any uplink or downlink message, which failed processing more times than the retry limit, will hold in Dead Letter Queue.

HARDWARE DETAILED DESIGN

In production environment, the following hardware design addresses on high scalability and high availability.

Six ActiveMQ service nodes are equally divided into two groups, and three ZooKeeper instances act as coordinators. The two groups of ActiveMQ service nodes share these ZooKeeper instances.

For each ActiveMQ service node in the two groups, we choose “LevelDB + ZooKeeper” as a hot standby solution to realize high availability¹³ (Fig. 3). We set the “replicas=2” to guarantee that only one ActiveMQ service node in each group is in “Master” state.¹⁴ In that case, two ActiveMQ service nodes will be in “Master” state simultaneously.

Setting all ActiveMQ service nodes to “Multicast Discovery” to realize high performance, we set the same multicast address among ActiveMQ service nodes (could use ActiveMQ’s default multicast address), so as to either listen for or advertise discovery events on a multicast address.¹⁵ So, these two ActiveMQ service nodes, in “Master” state, are

building a high-scalability solution (those “Slave” nodes will not send multicast messages).

BUSINESS RULES

As mentioned at above, small-scale telemedicine systems use provided software development kit to handle uplink and downlink message. Therefore, there is no need to be concerned with the handling of normalizing the uplink and downlink message format. By now, it should be clear that the most important business rule is how to plan queues of uplink and downlink message storage.

As Figure 4 illustrates that these joined small-scale telemedicine systems share the same uplink message queue, because they are secondary developed based on provided software development kit. Therefore, all uplink synchronization message’s format is consistent, and RTS uses one processing logic.

When downlink synchronization occurs, such as cross-area teleconsultation requesting or a new system joining RTS, it needs to send relative information to objective systems through downlink message queue. With existing multiple downlink message consumers (small-scale telemedicine systems), these consumers have their own internal processing logics, so there are multiple downlink message queues that relate appropriately to different consumers. In addition, when small-scale telemedicine system fails, downlink message will not be deleted until message has processed correctly.

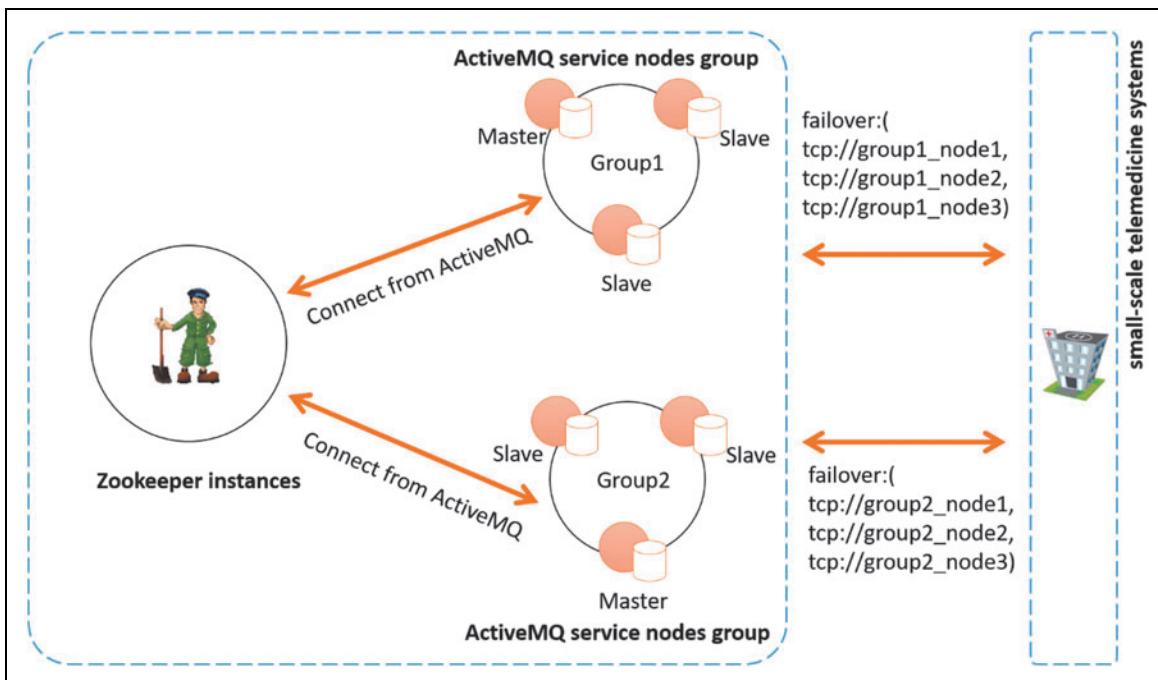


Fig. 3. Hardware design diagram of RTS. Six ActiveMQ service nodes equally divide into two groups, and three ZooKeeper instances act as coordinators. Two groups of ActiveMQ service nodes share these ZooKeeper instances. For each ActiveMQ service node in two groups, we choose “LevelDB + ZooKeeper” as hot standby solution to realize high availability. We set the “replicas=2” to guarantee that only one ActiveMQ service node in both groups is in “Master” state. Thus, this configuration allows one service node shutdown. Two “Master” state nodes provide service simultaneously.

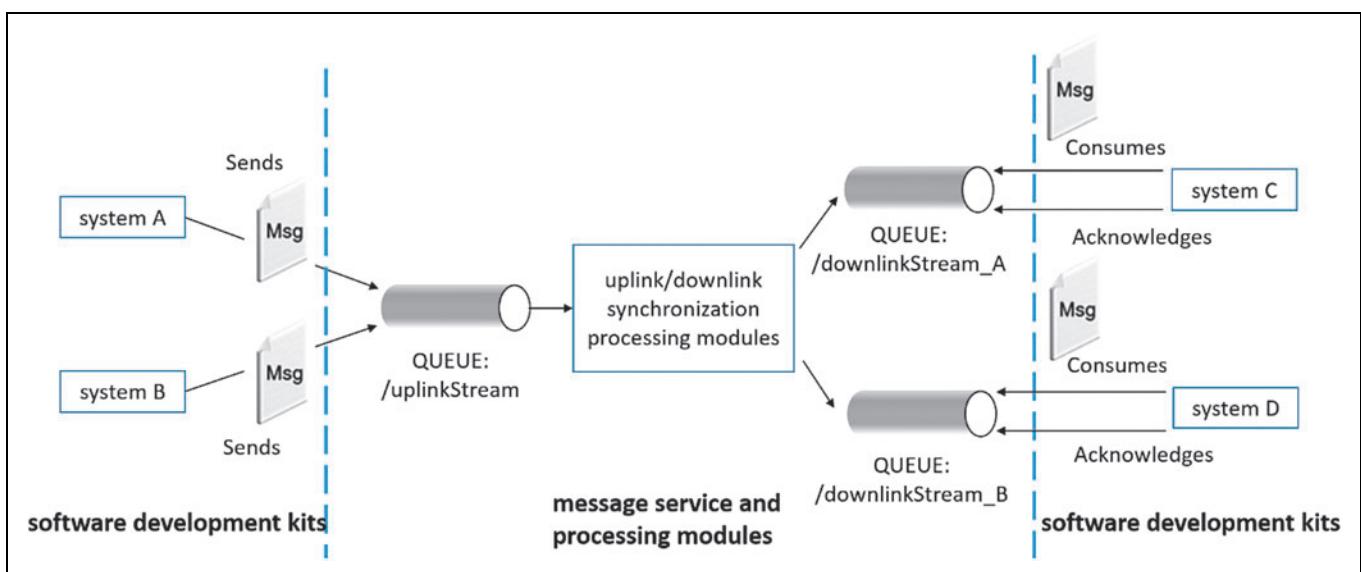


Fig. 4. Business rules of RTS. Small-scale telemedicine systems (such as system A–D) use software development kit, provided by RTS, to handle uplink and downlink message. These systems are secondary developed based on provided software development kit, therefore, all formats of uplink synchronization messages are consistent, and these systems could share the same uplink message queue. Provided kit encapsulates required information of patient as message, and sends message to the same uplink message queue. When downlink synchronization occurs, such as cross-area teleconsultation requesting or a new system joining RTS, it needs to send relative information to objective systems through downlink message queue. With existing multiple downlink message consumers (small-scale telemedicine systems), these consumers have their own internal processing logics, so there are multiple downlink message queues that relate appropriately to different consumers.

Table 1. Development Languages and Start Times of Eight Systems

SYSTEM	DEVELOPMENT LANGUAGE	SYSTEM START TIME
System_A	JAVA	2009
System_B	JAVA	2004
System_C	C#	2011
System_D	JAVA	2011
System_E	JAVA	2010
System_F	JAVA	2012
System_G	C#	2010
System_H	JAVA	2008

Results

Building RTS and receiving other systems took a total of 6 months, and eight small-scale telemedicine systems successively joined RTS. *Table 1* shows development languages and start times of eight systems. Eight systems were developed by means of mainstream development languages (TOP 10 Programming Language from TIOBE),¹⁶ and six out of eight systems were using JAVA. We developed RTS by using JAVA and provided Java and C# software development kits to facilitate communication.

After 13 months of the official launch of the RTS, system operation was stabilized without RTS breaking down, except for several service node failures. The number of ActiveMQ service node failure was recorded, with 53 cases between January and October 2015 (*Table 2*). Results show that only 28.3% (15 cases) were internal problems of RTS, the rest of 71.7% (38 cases) were caused by other objective reasons, such as network adjustment, patching operating system, network adjustment, or unknown server causes (such as high CPU utilization from unknown processes). From the 15 “Application Failure” cases, there are six cases caused by configuring ActiveMQ service nodes’ “transport connector configuration” to let nodes only support AMQP, which is an OASIS standard and a vendor-neutral and platform-agnostic protocol.^{17,18} There are six cases caused by configuring ActiveMQ service nodes’ “LevelDB,”

such as configuring log size and synchronization strategies. Three cases were caused by the unavailability of a web server.

Finally, teleconsultations of 13 months were recorded. A group of engineers instructed the related staff of small-scale telemedicine system about the use of joined system. After the first 2 months, the transitional period, the number of cases maintained around 1,125 in the following 8 months. In the last 3 months, shown at *Figure 5*, the number of relative cases increased.

Discussion

Telemedicine has reported as a useful health tool to improve healthcare outcomes,^{5,19} and telemedicine system in China has been evolved many times.²⁰ This article presents an efficient way to build extensible RTS based on message queue.

In this project, eight small-scale telemedicine systems, used in eight different cities, were involved. One RTS, used in one province, was developed. The RTS was designed to run 24 h per day, 7 days per week, and the RTS had permanent support from several engineers. The system, implementing RTS and involving eight small-scale telemedicine systems, costs 6 months.

The system has high availability, decreasing the possibility of service unavailability. Although 53 cases of ActiveMQ service node failure occurred in 13 months, we ensured that the system did not break down; ZooKeeper was used to guarantee that at least one node went well in each group. Moreover, when the failed service node recovered, it would join the former group automatically, without the need for a restarting of the whole group. It is worthy to note that the storage would synchronize from the healthy node to the recovery node behind the scenes. Hence, the use of “LevelDB + ZooKeeper” gives RTS high availability.

The system also has high scalability. As noted in the Hardware Detailed Design section, there are two ActiveMQ service nodes in “Master” state. In other words, two nodes offer services as soon as the “Multicast Discovery” is used. More services nodes could be added to the service group without service shutting down when necessary.

On the contrary, one of the RTS challenges is data consistency, especially when a failed service node recovered. In this project, although we chose “LevelDB + ZooKeeper” to ensure

Table 2. Number of Failures of Different Types of ActiveMQ Service Node

APPLICATION FAILURE	NETWORK ADJUSTMENT	STORAGE ADJUSTMENT	PATCHING OPERATING SYSTEM	DATA CENTER ADJUSTMENT	UNKNOWN SERVER CAUSES
15	7	8	5	4	14

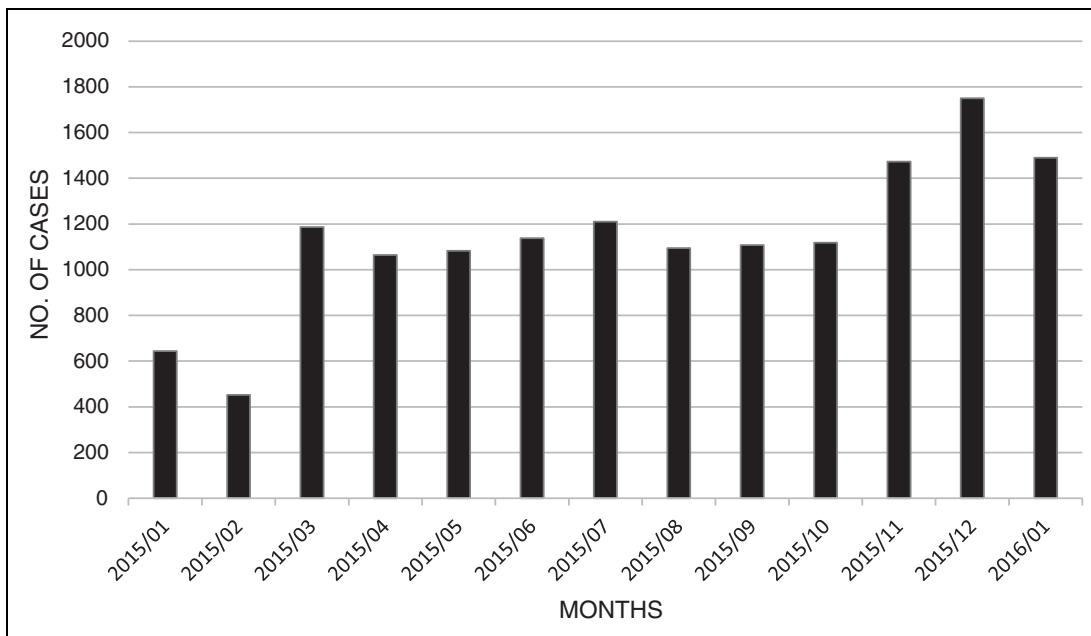


Fig. 5. Teleconsultation number over 13 months.

data consistency among service nodes, synchronize storage between nodes was an activity of resource consuming. Moreover, it might affect business services. For this reason, the next step is to use shared storage in ActiveMQ service group to separate business and storage logic.

Finally, although there have been several RCMSs and RCMS that shared medical resources, it really costs much time and money to build more RCMS²⁰ and there still have small-scale telemedicine systems running independently. This article has shown the potential of using message-based technology to building a high availability and scalability RTS.

Disclosure Statement

No competing financial interests exist.

REFERENCES

1. NHS Executive. *Information for health: An information strategy for the modern NHS 1998–2005: A national strategy for local implementation*. London, United Kingdom: NHS Executive, 1998.
2. World Health Organization. *The world health report: 1998: Life in the 21st century: A vision for all: Executive summary*. Geneva, Switzerland: World Health Organization, 1998.
3. Roine R, Ohinmaa A, Hailey D. Assessing telemedicine: A systematic review of the literature. *CMAJ* 2001;165:765–771.
4. Perednia DA, Allen A. Telemedicine technology and clinical applications. *JAMA* 1995;273:483–488.
5. Craig J, Petterson V. Introduction to the practice of telemedicine. *J Telemed Telecare* 2005;11:3–9.
6. Urtubey X, Petrich M. Argentina's national telemedicine programme: Reasons for a premature failure. *J Telemed Telecare* 2002;8(Suppl 3):69–71.
7. Cifuentes C, Romero E, Godoy J. Design and implementation of a telepediatric primary-level and low-cost system to reduce unnecessary patient transfers. *Telemed J E Health* 2017;23:521–526.
8. Wang Z, Gu H. A review of telemedicine in China. *J Telemed Telecare* 2009;15:23–27.
9. Zhao J, et al. Development and recent achievements of telemedicine in China. *Telemed J E Health* 2010;16:634–638.
10. Zhao J. Electronic health in china: From digital hospital to regional collaborative healthcare. *Information on Technology and Application in Biomedicine, 2008. ITAB 2008. International Conference on May 30–31 2008 in Shenzhen, China: IEEE, 2008* [Epub ahead of print]; DOI: 10.1109/ITAB.2008.4570514.
11. Zhai Y, et al. Clinical-and cost-effectiveness of telemedicine in type 2 diabetes mellitus: A systematic review and meta-analysis. *Medicine* 2014;93:e312.
12. Snyder B, Bosnanac D, Davies R. *ActiveMQ in action*. Vol. 47. New York, New York: Manning Greenwich Conn., 2011.
13. ActiveMQ. Introduction to master/slave. Available at <http://activemq.apache.org/masterslave.html> (last accessed February 20, 2018).
14. Junqueira F, Reed B. *ZooKeeper: Distributed process coordination*. Sebastopol, California: O'Reilly Media, Inc., 2013.
15. ActiveMQ. What is the difference between discovery, multicast and zeroconf. Available at <http://activemq.apache.org/what-is-the-difference-between-discovery-multicast-and-zeroconf.html>
16. TIOBE. TIOBE index. Available at <https://tiobe.com/tiobe-index/> (last accessed on February 20, 2018).
17. ActiveMQ. ActiveMQ supports the AMQP 1.0 protocol. Available at <http://activemq.apache.org/amqp.html> (last accessed on February 20, 2018).
18. OASIS. OASIS advanced message queuing protocol (AMQP) TC. Available at https://oasis-open.org/committees/tc_home.php?wg_abbrev=amqp (last accessed on February 20, 2018).
19. World Health Organization. *Telemedicine: Opportunities and developments in member States—a report on the second global survey on eHealth, Global observatory for eHealth series*. Geneva, Switzerland: World Health Organization, 2010.

20. Zhao J, et al. E-health in China: Challenges, initial directions, and experience.
Telemed J E Health 2010;16:344-349.

Address correspondence to:

Yuhong Li

Department of Information

The First Affiliated Hospital of Zhengzhou University
No 1 East Jianshe Road
Zhengzhou 450052
People's Republic of China

E-mail: regionalmedco@163.com

Yunkai Zhai

Department of Telemedicine

The First Affiliated Hospital of Zhengzhou University
No 1 East Jianshe Road
Zhengzhou 450052
People's Republic of China

E-mail: 1244913296@qq.com

Received: February 22, 2018

Revised: March 24, 2018

Accepted: March 24, 2018

Online Publication Date: May 31, 2018