Prototyping SMS Services for Medical Prescription Adherence

A. Ejnioui[1], M. Morjaret[2], and C. E. Otero[1]
[1]Information Technology, University of South Florida Lakeland, Lakeland, Florida, USA
[2]Computer Science and Networks, ESISAR, Valence, France

Abstract – *Barriers to medication adherence among patients have been shown to have significant impact on service quality and cost in the healthcare system. To minimize this impact, many in the healthcare industry are highly interested in supporting prescription adherence among patients. They believe that information technology in general, and mobile technology in particular, can help in developing medical practices that can be highly conducive to high rates of prescription adherence by enhancing communication between patients and healthcare providers. To this end, a number of pharmacy management benefit companies plan to adopt SMS communication to reach their customers given the wider acceptance of SMS messaging among mobile phone users. However, most of these pharmacies are reluctant to purchase service agreements from SMS aggregators without a complete understanding of user, service and business requirements related to SMS messaging. Hence, many are in dire needs for prototypes of SMS servers that can help them define and refine these requirements before committing to costly agreements with SMS aggregators. This paper describes such a prototype for a pharmacy benefit management company located in the southeast of the United States.*

Keywords: Mobile technology, Prescription adherence, Short message service, SMS aggregator, Pharmacy Services.

1 Introduction

Adherence is defined as the extent to which patients take medications as prescribed by their healthcare providers. A 2001 survey showed that although 62% of physician office visits generate a prescription, these prescriptions are not always adhered to [1, 2]. Poor adherence tends to be serious among patients who suffer from chronic diseases since these diseases require long-term treatments (e.g., HIV infections, hypertension, asthma, diabetes, heart disease and psychiatric illness). This is even more critical considering that 75% of all health expenditures in 2000 went to care for individuals with chronic illnesses although these individuals represent only 45% of Americans [3, 4]. In fact, non-adherence to prescribed medication is responsible for 10% of hospital admissions and 25% of nursing home admissions. It is estimated that the healthcare system in the U.S. incurs a cost of $300 billion annually due to non-adherence to essential medications. Patient surveys about non-adherence reveal an array of barriers to adherence such as costs of drugs, forgetfulness (e.g., it is practically difficult for a patient to remember to take medication several times a day), lack of clarity in the purpose of treatment, perceived lack of medication effect, debilitating side effects (e.g., for some professionals such as doctors, lawyers, professors and writers, the side effect of taking anticonvulsant drugs can interfere severely with abstract thinking), complicated regiment, lack of clarity in administration instructions, physical difficulty in handling medication (e.g., opening containers, handling small tablets), and unattractive formulation (e.g., unpleasant taste). According to the World Health Organization, increasing the effectiveness of adherence interventions may have a far greater impact on the health of world populations that any improvement in specific medical treatments [5]. For healthcare providers such as hospitals and insurance companies, strong adherence can lead to improved performance, which in turn can generate financial incentives for these providers. Providers can use improved performance as a metric to determine whether their services meet the expectation of their customers or not. Today, most people own a cell phone. It is conceivable to design mobile applications with user-friendly interfaces to help interested users in restoring their good health. Considering the current advantages of mobile technology and its communication facilities, it is clear that this technology can be exploited to help people learn to live healthy. In addition, it can be used effectively to personalize the therapy offered to a given individual considering his/her needs. To do so, a mobile application can be readily conceived as a medication adherence management tool on the go for the patient.

2 Prescription Adherence in Managing Pharmacy Benefits

For pharmacies, strong adherence can lead to a volume increase in prescriptions refills as well as access to patients who were otherwise invisible to drug manufacturers for marketing promotions. Of special note is the importance of employers and pharmacies in augmenting adherence if both stakeholders collaborate in designing smart pharmacy benefits. These benefits can increase prescription use without impacting overall drug expenditures in the healthcare system. Most pharmacy benefit management companies prefer to use Short Message Service (SMS) messaging to communicate with their customers considering its simplicity and wider acceptance [6]. However, these companies lack a suitable

infrastructure of information technology to do so. They can solicit the services provided by SMS aggregators by negotiating a cost-effective service level agreement with these aggregators that meets the requirements of SMS communication between the pharmacy and its customers. Worse yet, most pharmacies do not know what requirements must be taken in consideration to insure a successful message service with their customers. These requirements can be related to user interaction with the service, characteristics of the message service, and requirements related to business criteria as shown in Table 1.

Table 1. SMS service requirements.

Category	Requirement
User	• Number of messages per hour or day • Appropriate delivery time of messages (before midnight) • User responsiveness to messages • Sequence of messages in prescription adherence scripts • Suitability of interaction with prescription script messages
Service	• Message sending (batch, number of retries, queuing, etc.…) • Retrieving messages • Checking delivery of message status • Error and exception handling • Logging and tracking • Service configurability • Data storage (messages, customers, message traffic, etc.…)
Business	• Number of short codes • Provisioning of short codes • Type of network connections to the SMS gateway • Transactions per day • Transactions per second • Message content

In the absence of well-defined requirements, a pharmacy benefit management company might make its best effort to purchase a service package with an aggregator only to realize later that the purchased package does not satisfy the requirements of its SMS communication with its customers. There is always a risk of over- or under-shopping for these service packages. Hence, it becomes reasonable for such a company to develop a prototype of an SMS service in order to define and refine these requirements. Such a prototype can be used as an exploratory tool for developing requirements that can be used as guidelines to purchase the most suitable service package from an aggregator. In this context, a pharmacy benefit management company located in Florida, USA decided to build an SMS server prototype to generate such requirements. This paper describes the architecture and design of the prototype of this SMS server.

3 SMS Service Architecture

Although the pharmacy benefit management company mentioned above did not have a complete understanding of requirements in each category, it opted to base the design of the message server on the following requirements:

- *Self-Containment*: The message server must contain all the computing resources it needs to separate its responsibilities from those of the software applications of the company.
- *Logging*: This capability is needed to keep track of various events taking place between the software applications and the message server. The purpose of this tracking is to help the company determine the most important user, service and business requirements.
- *Error Handling*: This capability is need to record all errors and exceptions between the SMS gateway server of the network service provider and the software applications of the company. Recording these errors can provide a rich perspective on the reliability of the service offered by the network service provider.
- *Configurability*: This capability allows the benefit management company to manage the message service in different ways in order to explore requirements that are not clearly understood in normal operating conditions of the message server.

Based on these initial requirements, the architecture of the message server has been developed and refined over several iterations to include the following components as shown in Figure 1:

- *Front Interface*: This interface provides methods that can be called by the software applications of the pharmacy benefit management company to perform tasks related to communication with its customers via SMS messages.
- *Message Database*: This database is a persistent store sued to record sent messages, received messages, errors and exceptions generated during SMS message exchange between the software applications and customer cell phone.
- *Message Server*: This server is a process that runs continuously to record all the events related to SMS communication between the software applications the customers such as sending message, retrieving reply messages and checking the delivery status of sent messages.
- *Data Access Layer*: This layer consists of dynamic libraries responsible for passing data from and to the message database on behalf of the front interface, the message server and the back interface.
- *Back Interface*: This interface provides methods to the message server for sending messages, retrieving messages, and checking the delivery status of sent message via the application programming interface (API) of the SMS gateway of the network service provider [7].

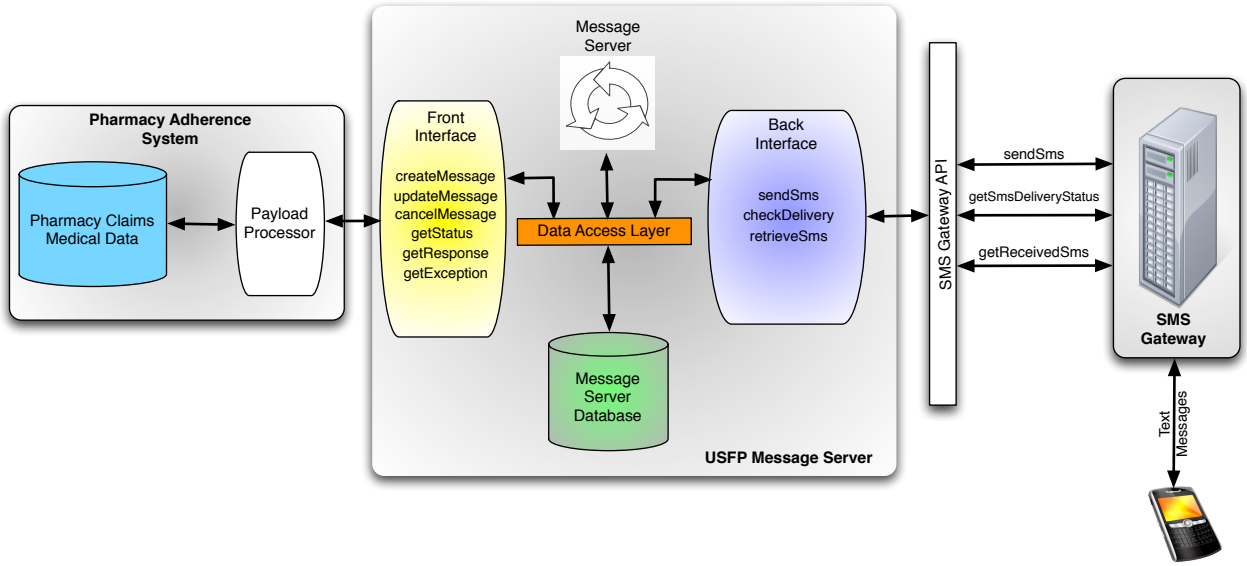


Figure 1. Architecture of the message server.

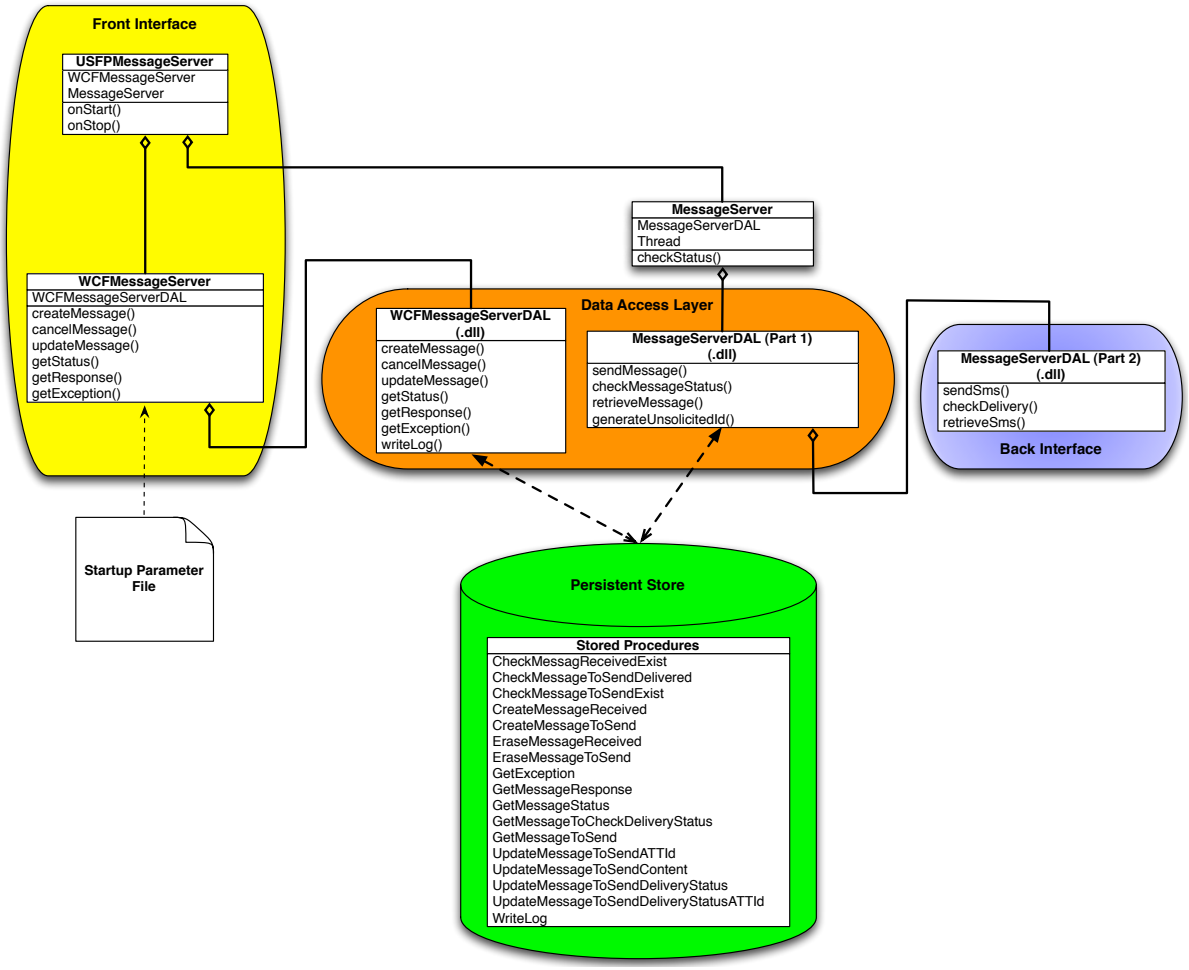


Figure 2. Class mapping on the architecture of the message server.

4 SMS Server Implementation

The architecture shown in Figure 1 was used to derive a class hierarchy for implementing the components seen in the architecture. This hierarchy was implemented using C# on .NET [8]. Figure 2 shows the classes of the message server.

The Front Interface
The front interface consists of the WCFMessageServer and MessageServer classes. Table 2 summarizes the methods of these two classes.

The Message Server
The message server consists of the MessageServer class. This class spawns a thread responsible for calling repetitively the sendMessage, checkMessageDelivery and retrieveMessage methods in the MessageServerDAL class. These repetitive calls are performed as long as the number of transactions does not exceeded the maximum number of transactions allowed per day by the network service provider. Most network service providers impose limits on the number of transactions completed between a company server and their own SMS gateways based on the service level agreement purchased by the company in need of SMS services. It is meant by a transaction any call to the SMS gateway server of the network service provider.

The Data Access Layer
The data access layer consists of the WCFMessageServerDAL and the first part of the MessageServerDAL classes. Table 3 summarizes the methods of these two classes.

The Message Database
The message database consists of the following tables:

- Table of messages to send: This table contains records of the messages that need to be sent to the SMS gateway.
- Table of received messages: This table contains records of received messages. These messages are reply messages sent by customers are replies to messages sent by the pharmacy benefit management company.
- Message table: This table contains records of messages generated for events that took place while the message server is in operation. These events can be errors, exception or log entries recording specific tasks completed by one of the classes in the message server.

In addition to these tables, this database stores a number of procedures shown in Figure 2.

The Startup Parameter File
When the message server starts, it needs to upload several parameters for proper functionality. These parameters are:

- Database connection settings: These are the settings necessary for the server to establish the connection with the database. They consist of the location path of the database and its security settings.
- Short code: This is the code assigned by the network service provider to the customer. This code is used to address SMS messages coming to or leaving from the servers of the pharmacy benefit management company.
- Endpoint send address: This is the URL address to which messages must be sent as required in the SMS gateway API.
- Endpoint receive address: This is the URL address to which message must be received as required in the SMS gateway API.
- Number of transactions per day: This number is fixed by the network service provider based on the service level agreement purchased by the pharmacy benefit management company.
- Number of transactions per second: This number is fixed by the network service provider based on the service level agreement purchased by the pharmacy benefit management company
- Transaction counter: This is a software counter generally initialized to 0 unless specified otherwise at startup time.

These parameters are stored in a file that is used by the message server during startup to load these parameters.

5 SMS Communication via the Message Server

Communication between the pharmacy benefit management company and its customers intended to enforce prescription adherence mostly of scripted dialogs between the company and its customers. The dialog below illustrates a simple adherence communication session between the pharmacy and a customer named DuPont.

Sending A Reminder Message To The Customer
In the first step, the pharmacy sends a message to Mr. DuPont to remind him to take his medication as shown in Figure 3. Before the message is forwarded to the SMS gateway, it is inserted in the tables of messages to send in the database. Figure 4 shows that the first entry is the entry of this message in the table. This entry shows that this message has reached the cell phone of Mr. DuPont since its delivery status has been automatically updated to 'DeliveredToTerminal'.

Receiving A Reply From The Customer
After customer DuPont receives the reminder message, he replies affirmatively by sending a "Yes, I did." reply message as shown in Figure 5. As soon as the message server receives a this reply, the reply is immediately inserted in the table of received messages as shown in Figure 6. The reply is passed back to the software applications of the pharmacy.

Sending An Acknowledgement To The Customer
When the pharmacy receives the reply message, its script dictates that it sends an acknowledgment to the customer as

Table 2. Front interface classes and methods.

Class	Method	Description
USFPMessageServer	onStart	It starts the Windows service.
	onStop	It stops the Windows service.
WCFMessageServer	createMessage	It calls the createMessage method in the WCFMessageServerDAL class by passing a message object.
	cancelMessage	It calls the cancelMessage method in the WCFMesssageServerDAL class by passing a message id.
	updateMessage	It calls the udpateMessage method in the WCFMessageServerDAL class by passing a message object.
	getStatus	It calls the getStatus method in the WCFMessageServerDAL class by passing a message id.
	getResponse	It calls the getReponse method in the WCFMessageServerDAL class by passing a message id.
	getException	It calls the getException method in the WCFMessageServerDAL class by passing a start and end dates.

Table 3. Classes and methods of the data access layer and back interface.

Class	Method	Description
WCFMessageServerDAL	createMessage	It calls the CreateMessageToSend stored procedure to insert each message in a batch of messages if the message does not already exist in the table of messages to send in the database.
	cancelMessage	It calls the EreaseMessageToSend stored procedure to remove the message from the table of messages to send and EraseMessageReceived stored procedures to remove the message from the table of received messages.
	updateMessage	It calls the UpdateMessageToSendContent stored procedure to update the message contents in the table of messages to send in the database.
	getStatus	It calls the getStatus stored procedure to obtain the status of a sent message from the table of message to send in the database.
	getResponse	It calls the getMessageResponse stored procedure to extract the reply messages from the table of received messages.
	getException	It calls the GetException stored procedure to extract exceptions between two timestamps from the table of exceptions.
	writeLog	It class the WriteLog stored procedure to write relevant information to the messages table about an event taking place while the message server is in operation.
MessageServerDAL (Part 1)	sendMessage	It extracts the messages that need to be sent from the table of messages to send and calls the sendSms method in the MessageServerDAL class.
	checkMessageStatus	It calls the checkDelivery method in the MessageServerDAL class for each message whose status needs to be checked from the table of message to send.
	retrieveMessage	It calls the retrieveSms method in the MessageServerDAL class to retrieve reply messages from the SMS gateway.
MessageServerDAL (Part 2)	sendSMs	It creates a connection to the SMS gateway and calls the sendSms method in the SMS gateway API in order to send a batch of messages.
	checkDelivery	It creates a connection to the SMS gateway and calls the getSmsDeliveryStatus method in the SMS gateway API in order to check the delivery status of a batch of sent messages.
	retrieveSms	It creates a connection to the SMS gateway and calls the getReceivedSms method in the SMS gateway API in order to retrieve a batch of reply messages.

shown in Figure 7. Before the message is forwarded to the SMS gateway, it is inserted in the tables of messages to send. Figure 8 shows that the second entry is the entry of this acknowledgement message. This entry shows that this message has reached the cell phone of Mr. DuPont since its delivery status has been automatically updated to 'DeliveredToTerminal'

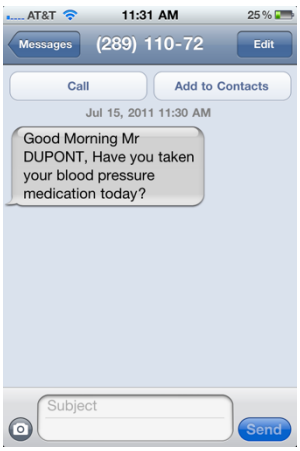
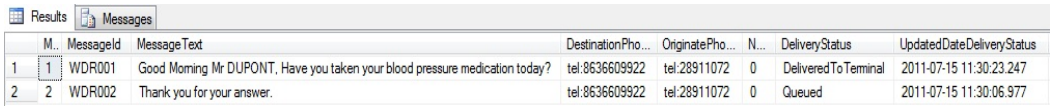


Figure 3. Reminder message to Mr. DuPont.

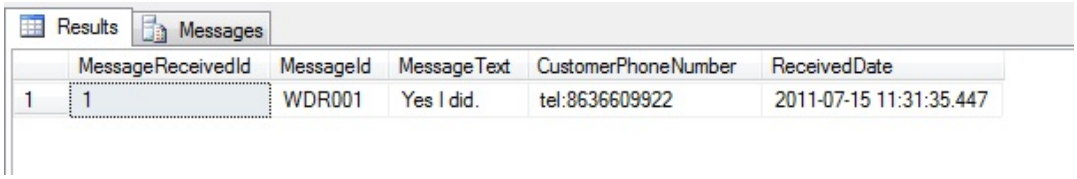
Figure 5. Rely message from Mr. DuPont.

Figure 7. Acknowledgement message from the pharmacy.



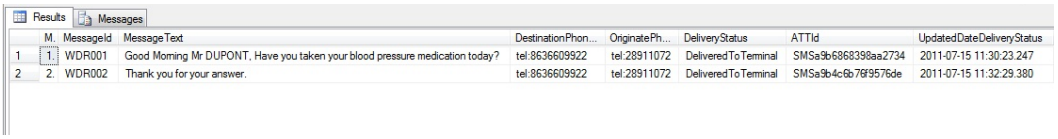
	M..	MessageId	MessageText	DestinationPho...	OriginatePho...	N...	DeliveryStatus	UpdatedDateDeliveryStatus
1	1	WDR001	Good Morning Mr DUPONT, Have you taken your blood pressure medication today?	tel:8636609922	tel:28911072	0	DeliveredToTerminal	2011-07-15 11:30:23.247
2	2	WDR002	Thank you for your answer.	tel:8636609922	tel:28911072	0	Queued	2011-07-15 11:30:06.977

Figure 4. Contents of the table of messages to send.



	MessageReceivedId	MessageId	MessageText	CustomerPhoneNumber	ReceivedDate
1	1	WDR001	Yes I did.	tel:8636609922	2011-07-15 11:31:35.447

Figure 6. Contents of the table of received messages.



	M..	MessageId	MessageText	DestinationPhon...	OriginatePh...	DeliveryStatus	ATTId	UpdatedDateDeliveryStatus
1	1.	WDR001	Good Morning Mr DUPONT, Have you taken your blood pressure medication today?	tel:8636609922	tel:28911072	DeliveredToTerminal	SMSa9b6868398aa2734	2011-07-15 11:30:23.247
2	2.	WDR002	Thank you for your answer.	tel:8636609922	tel:28911072	DeliveredToTerminal	SMSa9b4c6b76f9576de	2011-07-15 11:32:29.380

Fig. 8. Contents of the table of message to send.

6 Conclusion

This paper presented the prototype architecture and implementation of an SMS server intended to help a pharmacy benefit management company to define its SMS service requirements. These requirements can be used to shop for a service level agreement from an SMS aggregator that is highly suitable to the needs of the benefit management company.

7 References

[1] D. K. Cherry, C. W. Burt, D. A. Woodwell, "National Ambulatory Medical Care Survey," *Advanced Data from Vital Statistics*, no. 358, 2005, pp. 1-40.

[2] Boston Consulting Group and Harris Interactive, "The Hidden Epidemic: Finding a Cure for Unfilled Prescriptions and Missed Doses," December 2003, available at http://www.bcg.com/publications/files/TheHiddenEpidemic_Rpt_HCDec03.pdf.

[3] G. Anderson, J. Krickman, "Changing The Chronic Care System to Meet People's Needs," *Health Affairs*, vol. 20, no. 6, 2001, pp. 146-160.
[4] C. Hoffman, D. Rice, H.-Y. Sung, "Persons with Chronic Conditions: Their Prevalence and Costs," *Journal of American Medical Association*, vol. 276, no. 18, 1996, pp. 1473-1479.
[5] World Health Organization, "Adherence to Long-Term Therapies: Evidence for Action," 2003, available at http://www.who.int/chronic_conditions/en/adherence_report.pdf .
[6] Y. Mao, Y. Zhang, S. Zhai, "Mobile Phone Text Messaging for Pharmaceutical Care in a Hospital in China," *Journal of Telemedecine and Telecare*, vol. 14, no. 8, 2008, pp. 410-414.
[7] AT&T Network Services, *Network Services API Developer Guide*, AT&T, 2010.
[8] Microsoft, "Introduction to Windows Service Applications," Microsoft Developer Network, 2005, available at http://msdn.microsoft.com/enus/library/d56de412(VS.80).asp.