Implementation of Parlay X interfaces for standard GSM modems to support the development of SMS-based applications

Enio Kaljić*, Mesud Hadžialić* & Muhamed Kurić**
Department of Telecommunication*
Department of Automatic Control and Electronics**
Faculty of Electrical Engineering at the University of Sarajevo
Zmaja od Bosne, 71000 Sarajevo, Bosnia and Herzegovina
Email: ekaljic@etf.unsa.ba, mhadzialic@etf.unsa.ba, muhamed.kuric@etf.unsa.ba

***Abstract*—The goal of this paper was to develop a Simple Access Object Protocol (SOAP) Parlay X Web service interface that supports standard GSM modems and can be used as development and testing environment for SMS based applications. The service interface was translated into Java classes from a selected set of Parlay X version 2.1 Web Service Description Language (WSDL) files providing mandatory SMS functionality. SendSms, ReceiveSms, NotificationManager and SmsNotification Web services were fully implemented. Other Parlay X services (e. g. MMS, ringtone services and other) can be built in a similar fashion and easily integrated into the system architecture. The service backend was implemented using Java wrapper classes that mimic the described Web services, their datatypes and membership functions. The SIM300 GSM modem was used to transmit and receive SMS, while a MySQL database schema was used as interface between the modem and the service interface. Handler classes (also implemented in Java) were used as modem behaviour controller, while the database connection was established using the Java Database Connectivity classes provided by Oracle. Detailed System architecture, implemented Web services functionality, database structure and Message Sequence Charts are given.**

Keywords — Parlay, Parlay X, SMS, SOAP, SMS gateway, SIM300, WSDL.

I. INTRODUCTION

With the emerge of Next generation networks (NGN), Service oriented architecture has become the dominant software architecture in telecommunicational software engineering. Web service frameworks usually provide an emulator which gives software developers a simulation platform to test the developed Web services. Those services usually depend on existing telecommunicational services (e.g. SMS and MMS services) defined by their respective standard. In order to test such services with real world data it is necessary to embed external communication functionality into the emulator.

As is shown in [1] the current trend on service provision in telecoms is based on open interfaces and protocols exposing core network functions. The use of standardized interfaces to interract with the network is a significant factor contributing transparency, modularity, reusability and clear distribution of administrative responsibilities to the service development process. In the paper [2] the Parlay X approach was adopted for exposing telecom capabilities as well as notifications of telecom-related information from the IMS (IP Multimedia Subsystem) network to the Web 2.0 domain. The telecom industry is gravitating towards a standard definition of a service delivery platform, which embraces telecom Web Services coupled with partner management, as an important element in service delivery [3].

The paper is organized as follows. A short overview of related work is presented in section 2. In section 3 the system architecture is described. Detailed information about the database schema is given in section 4. The implemented Parlay X SMS Web services are described in section 5. Conclusions and further research guidelines are discussed in the final section.

II. RELATED WORK

Mobile devices have limited resources and low bandwidth. Due to cellular data characteristics, interacting with Web Services from a mobile phone is problematic using plain HTTP. In order to increase the performance of mobile devices during the access of Web Services, telecommunication services are used. Therefore, the paper [4] describes Web services with SOAP binding for SMS protocol utilization.

In papers [5], [6], [7] the authors have made an integration of GSM modem and software solutions developed according to their needs, but in fact they did not observe a single standard for this type of integration, which made their solution inflexible, non-modular and difficult to administer.

This paper proposes a design to create a software system that provides mandatory SMS functionality with a standardized Parlay X SOAP Web service interface. The developed software implemented using the proposed approach can easily be integrated into existing Java based network simulators.

III. SYSTEM ARCHITECTURE

The system architecture is shown in Fig. 1. It consists of three layers: the service layer, the interface layer and the core layer. The service layer is based on the multitier service oriented architecture. SOAP Web services provide the Web interface to the external service consumers. The service interface translates Web service input data into database data and vice versa. The DataAccessObject (DAO) of the service layer is a object wrapper to the database. This way a classical delegate pattern is implemented via the shared database tables. The core layer handles the stored output data from the database and sends it to the receiving terminal units. When a new SMS is received it is stored via the core layer into the database and accessed by the relevant consumer via the respective Web service. The interface layer is used as middleware between the core and service layer and is used as asynchronous event notification from the core to the service layer. This way the notification service can be called from the core layer to notify subscribed clients.

IV. Database structure

The Entity relationship diagram (ERD diagram) of the database is shown in Fig. 2. The tables can be divided into two groups: inbox tables and outbox tables. Each message the modem receives is stored into the Inbox table. Each received message is matched against ClientList correlators provided by the service clients. The respective client is notified via his endpoint and interface name in case that the message matches his correlator. That way a software link between the Inbox and ClientList table exists. When a client invokes the Web service method named sendSms, the SMS message, as earlier explained, is first stored into the Outbox table and a unique delivery id is generated and returned to the client. The modem processes the message queue using the FIFO priority approach. A unique modem delivery id is generated for each message that the modem sends to the network. The modem delivery id and outbox table delivery id form a unique pair that is used for delivery status updates. The Client can send a SMS using optional fields providing message status tracking. Whenever a message status is updated via the network, the corresponding delivery id is matched with the registered delivery ids in the OutboxNotificationList. If a match is found then the respective client is notified via the provided endpoint and interface name.

V. Web services

The autogenerated Apache Tomcat list of hosted Web services and their respective methods is shown in Table I. Those whoare familiar with the Parlay X standard will recognize the Target namespace and the Endpoint address, as they are strictly predefined by the standard (As a side note, all Web service and method names are case sensitive). As we can see the ReceiveSms, SendSms and SmsNotificationManager Web services were implemented [8], [9].

To get better understanding of how each system component interacts during we will analyze Message Sequence Charts (MSC) for typical system scenarios.

The MSC of the getReceivedSms method from Table I is shown in Fig. 3. By invoking this method the user requests from the SMS Gateway all received short messages that

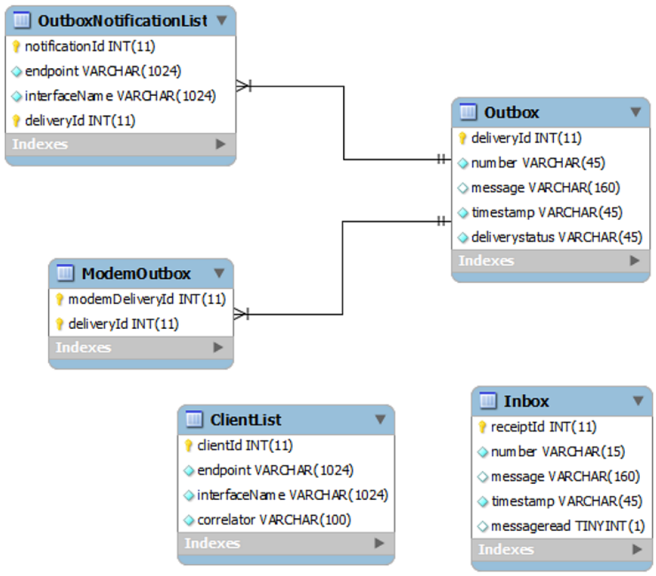

Fig. 2. Entity relationship diagram of the database schema

match the provided request correlator. This request is delegated into the database by calling the function DaoGetReceivedSms which returns the requested inbox message list.

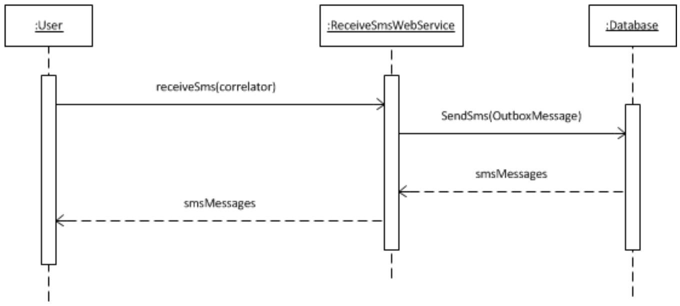

Fig. 3. MSC of the getReceivedSms method from the ReceiveSms web service

To be able to list received SMS we need to receive short messages. This scenario is shown in Fig. 4. The short messages arrive asynchrony from terminal devices and are stored into the modems' buffer. The Receiver class gets notified via a message tracking notification generated from the modem over serial port. When a new message arrives the Receiver class stores it into the database and deletes it from the modems' buffer.

The MSC of the sendSms method from Table I is shown in Fig. 5. By invoking this method the client requests from the SMS Gateway to send the short message provided in the request. This request is delegated into the database and the system creates a unique identifier for this message which is returned to the client. The Modem Core checks periodically if there is a short message for delivery in the database and delivers it to the client if such message exists. The network asynchrony returns status notifications of the delivery which the Modem Core updates in the database.

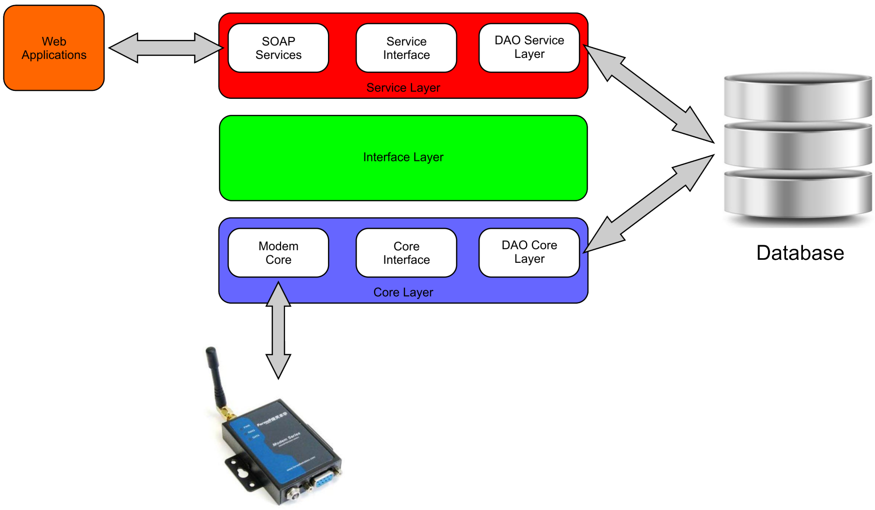

Fig. 1. SMS Gateway architecture

TABLE I
SMS GATEWAY AVAILABLE SOAP WEB SERVICES

ReceiveSms: • getReceivedSms	Endpoint address: http://146.185.20.25:8080/ParlayXSmsServices/services/ReceiveSms WSDL: {http://_interface.v2_1.receive.sms.parlayx.wsdl.csapi.org/}ReceiveSmsService Target namespace: http://_interface.v2_1.receive.sms.parlayx.wsdl.csapi.org/
SendSms: • sendSms • getSmsDeliveryStatus	Endpoint address: http://146.185.20.25:8080/ParlayXSmsServices/services/SendSms WSDL: {http://_interface.v2_1.send.sms.parlayx.wsdl.csapi.org/}SendSmsService Target namespace: http://_interface.v2_1.send.sms.parlayx.wsdl.csapi.org/
SmsNotificationManager: • startSmsNotification • stopSmsNotification	Endpoint address: http://146.185.20.25:8080/ParlayXSmsServices/services/SmsNotificationManager WSDL: {http://_interface.v2_2.notification_manager.sms.parlayx.wsdl.csapi.org/}SmsNotificationManagerService Target namespace: http://_interface.v2_2.notification_manager.sms.parlayx.wsdl.csapi.org/

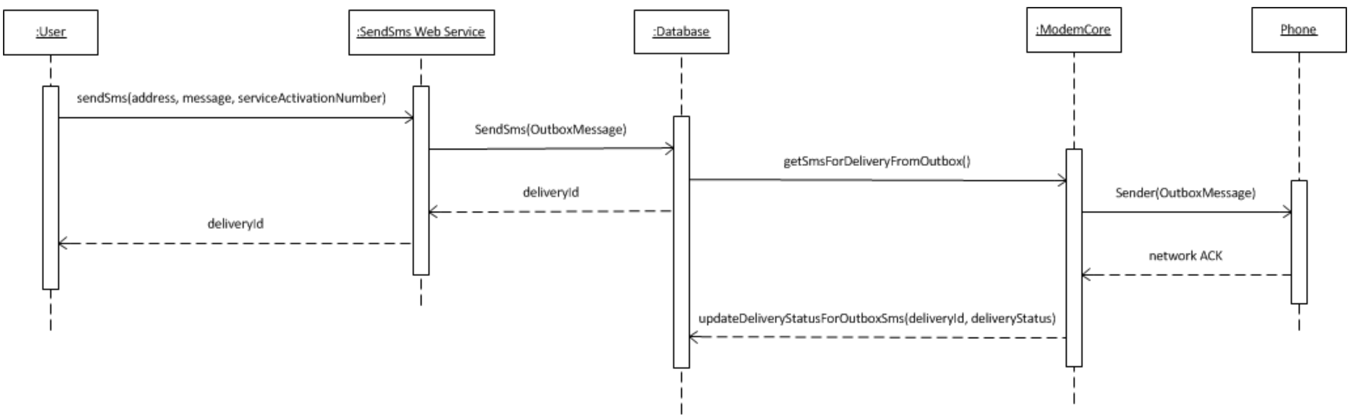

Fig. 5. MSC of the sendSms method from the SendSms Web service

The MSC of the getSmsDeliveryStatus method from Table I is shown in Fig. 6. The client passes the unique identifier he got when he sent a short message and the system returns the status of the message as defined in [8].

It is even possible to track the send short message by using the NotificationManager Web service. The MSC of the startSmsNotification and stopSmsNotificationMethods are shown in Fig. 7. and Fig. 8., respectively. The startSmsNotification method serves as subscription for message tracking, while stopSmsNotification cancels the active subscription.

The MSC of the sendSms method from the SendSms Web service with active short message tracking is shown in Fig. 9. The message flow is the same as in Fig. 5 with only difference that the client is receives status updates provided by the network. In order to be able to receive those notifications the client needs to implement the server side of the SmsNotification Web service given in [8], while the SMS gateway possess a client side implementation of this service.

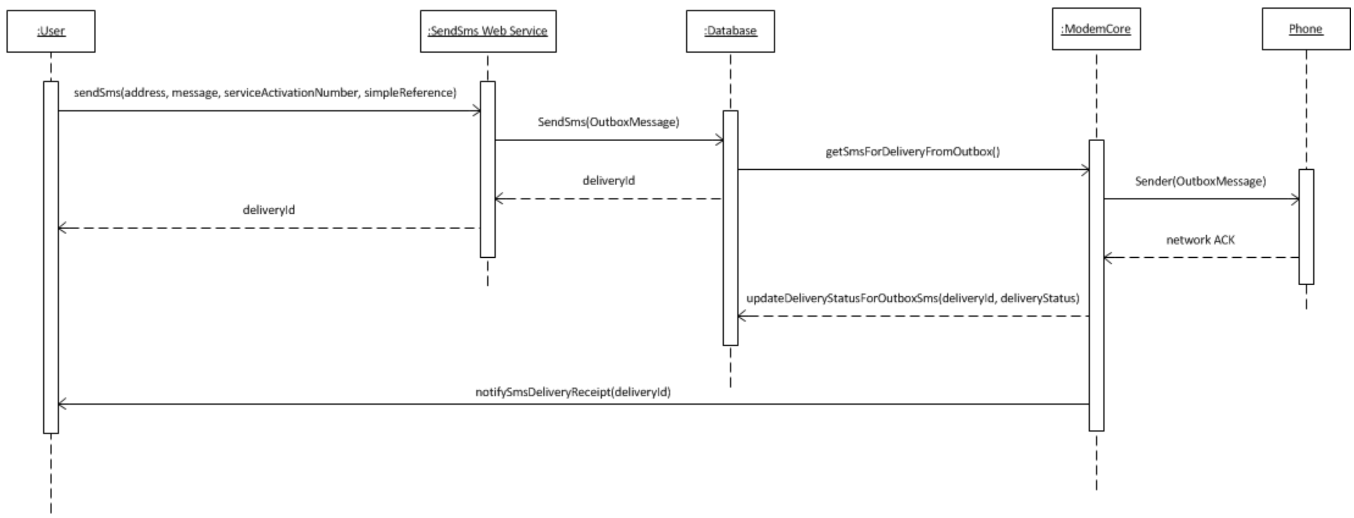

Fig. 9. MSC of the sendSms method from the SendSms Web service with message tracking

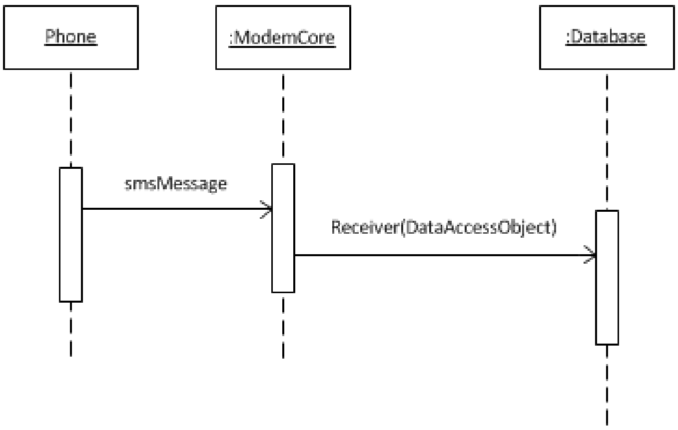

Fig. 4. SMS Gateway receiving SMS

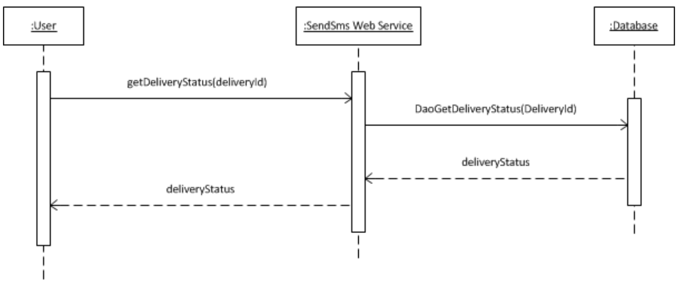

Fig. 6. MSC of the getSmsDeliveryStatus method from the SendSms Web service

VI. Conclusion

This paper presented a design approach to implement a SMS Gateway software. The proposed design is based on a stack multitier software architecture with SOAP Web service interface. The Web service interface was built from Parlay X version 2.1 Web service WSDL files. Detailed database ERD schema and implemented Web services MSC were given. The implemented software based on the proposed design can easily be integrated into a Network simulator. Other Parlay X functionalities, e. g. Call and MSM services, can easily be added which is result of the proposed layered architecture.

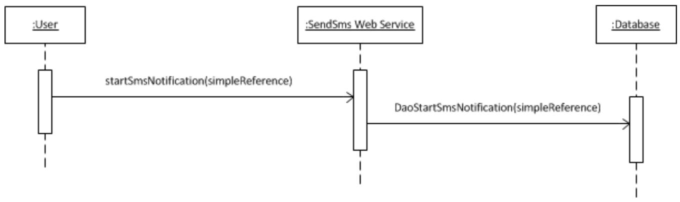

Fig. 7. MSC of the startSmsNotification method from the NotificationManager Web service

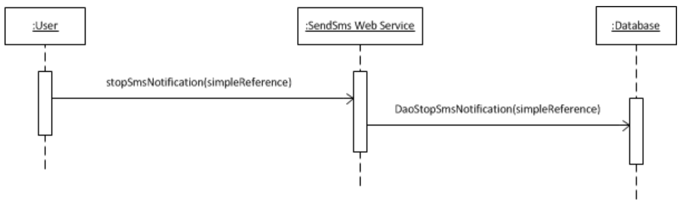

Fig. 8. MSC of the stopSmsNotification method from the NotificationManager Web service

Further research will cover the following topics:

- **Full Parlay X version 2.1 standard implementation:** As earlier explained only mandatory SMS functionality was provided in this implementation. The first step in the future would be to extend the system to provide full SMS functionality as per standard. After that other Parlay X defined Web services, e. g. Call and MSM services, should be implemented.
- **Emulator integration:** The primary goal of this paper was to provide a hardware software solution suited for real world data Web service testing. To achieve this goal it is necessary to find an emulator into which the implemented solution could be embedded.
- **SMS router:** The system design could be reused with

slight modifications to create a SMS router consisting of multiple SMS Gateways.

- **Modem compatibility:** In order to cover a vast range of modem models it is required to do further research on minimal modem AT command set that is necessary to implement all system requirements.

REFERENCES

[1] N. D. Tselikas, G. S. Tselikis & N. C. Sagias, Software and Middleware Technologies Based on Open APIs and Protocols for Modern Service Provision in Telecoms, pci, pp. 33-37, 2010 14th Panhellenic Conference on Informatics, 2010.

[2] J. M. E. Carlín & Y. B. D. Trinugroho, "A Flexible Platform for Provisioning Telco Services in Web 2.0 Environments," ngmast, pp. 61-66, 2010 Fourth International Conference on Next Generation Mobile Applications, Services and Technologies, 2010.

[3] E. W. Burger, S. Rajasekar, P. O'Doherty, A. Lundqvist & T. Grönberg, A telecommunications web services platform for third party network access and SOA-based service delivery, Proceedings of the 2007 Workshop on Middleware for next-generation converged networks and applications (MNCNA '07). ACM, New York, USA, 2007.

[4] R. Singh, S. Mishra & D. S. Kushwaha. 2009, An efficient asynchronous mobile web service framework, SIGSOFT Softw. Eng. Notes 34, December 2009.

[5] Z. Wang, Z. Gzo & Y. Wang, Design and Implementation of Short Message Query System for Academic Office Based-On J2ME, cccm, vol. 1, pp. 509-513, 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, 2008.

[6] Y. Wu, J. Chen & X. Ba, Application and Implementation of SMS Platform in Project Management Information System Approval Workflow, wcse, vol. 1, pp. 77-80, 2010 Second WRI World Congress on Software Engineering, 2010.

[7] B. Ramamurthy, S. Bhargavi & R. Shashi Kumar, Development of a Low-Cost GSM SMS-Based Humidity Remote Monitoring and Control system for Industrial Applications, International Journal of Advanced Computer Sciences and Applications, vol. 1, issue 4, 2010.

[8] The European Telecommunications Standards Institute, ETSI OSA Parlay X version 2.1 standard specification Part 4: Short Messaging (ETSI ES 202 391-4 V1.2.1), December 2006.

[9] The European Telecommunications Standards Institute, ETSI OSA Parlay X version 2.1 standard specification Part 1: Common (ETSI ES 202 391-1 V1.2.1), December 2006.