An Approach to Define Interfaces for Mobile Telemetry

Ivaylo Atanasov[1], Ventsislav Trifonov[1], Evelina Pencheva[1]

***Abstract* – The mobile telemetry has a lot of applications ranging from green technologies through telemedicine to security. The paper studies the generic functions of mobile telemetry to define an abstraction and to design Application Programming Interfaces (APIs). The mobile telemetry APIs for acquisition of measurements data are implementation protocol independent. The paper proposes Web Services access to measurement data which allows 3rd party applications to fetch data in secured and managed manner. Web Services access is also proposed for mobile telemetry system entities that provide data in the repository.**

***Keywords* – Application Programming Interfaces, Web Services, Database abstraction layer.**

I. INTRODUCTION

Mobile telemetry uses mobile data for remote measurement and reporting. The range of applications is from green technologies and security applications to telemedicine. The requirements to mobile telemetry protocols are currently subject of intensive research studies. Different aspects of telemetry functions and their implementations in mobile agents are discussed in [1], [2] and [3]. The main issues in the design of a mobile telemetry protocol concern the reliable data transfer and data security. Despite of the differences in application domains, any mobile telemetry protocol requires generic communication functions. We put the accent on the comparison between the generic functions of existing solutions. Townsend, Abawajy and Kim [4] present an approach to monitoring moving objects by utilizing mobile technologies and Short Message Service (SMS) to transmit data from a patient to the doctor. The SMS is pervasive messaging technology but it does not provide delivery guarantees and standardized delivery receipts. That is why the authors extend SMS with a message validity period, failover options and redundancies. However the proposed solution is not suitable for real-time applications because of unpredictable delays introduced by the SMS Centre. Cibuk and Balik [5] demonstrate a model-based solution approach for mobile telemetry. Their model is based on the communication medium of the Internet and transport layer in TCP/IP model. TCP is not suitable for real-time applications because it provides a reliable message transfer at the cost of retransmissions which implicitly increase the transport delay. Further, as application data may be sensitive, it needs to be transferred in a secure way, but the proposed protocol does not include any security services. In this paper, we suggest an approach to design Application Programming Interfaces (APIs) for mobile telemetry. The APIs define an abstraction of mobile telemetry functions and are independent of the implementation protocol. The API may be used in different mobile telemetry application domain.

Another aspect of mobile telemetry is the external access to measurements data. Traditionally, all database vendors provide their own interface tailored to their products which leaves it to the application developer to implement the code for the database interfaces. Information abstraction layers reduce the amount of work by providing a consistent API to the developer and hide the database specifics behind this interface as much as possible. An effective information encapsulation layer provides benefits such as coupling between the object schema and data schema, an ability to evolve either one, provisioning of a common place to implement data-oriented business rules, and increasing application performance [6]. There are a number of approaches that industries have employed when they want to expose their databases as Web Services. Abstraction layers with different interfaces in numerous programming languages such as Drupal 7 Database API, Java API for XML Web Services, SOAP with Attachments API for Java, Java DB, Java Data Objects, and NewsKnowledge Web Service API provide a standard, vendor-agnostic abstraction layer for accessing database servers. In parallel, intensive research is conducted on applying advanced technologies for the information abstraction layer. Jayasinghe [7] suggests an approach for exposing a database as a Web Service using Axis2. Yang, Zhang, and Zhao [8] analyze database connection mechanism of a WEB application system based on a three-tier architecture, and the corresponding relationship between main steps and auto-generated code used to achieve database access in the Dreamweaver development environment. Qu, Feng and Sun [9] analyze the difficulty of the united access of the distributed and heterogeneous biological information database and present an approach based on web service and multi-agent, which adds intelligence to the united access. Sellis, Skoutas and Staikos [10] describe efforts towards database integration and interoperability, based on Web Services and ontologies. In this paper, we define Web Services interfaces that provide an abstraction of the access to database storing mobile telemetry measurements.

The paper is structured as follows. Section II describes generic architecture for mobile telemetry and identifies common functions. Section III presents an approach to the definition of APIs for mobile telemetry. Section IV describes an approach to design Web Services interfaces for open but secured access to measurements database. Section V presents an example of Web Services usage. The conclusion summarizes the contributions.

[1]The authors are with the Faculty of Telecommunications, Technical University of Sofia, Kl. Ohridski 8, 1000 Sofia, Bulgaria, E-mails: iia@tu-sofia.bg; vgt@tu-sofia.bg; enp@tu-sofia.bg

II. GENERIC ARCHITECTURE FOR MOBILE TELEMETRY

The mobile telemetry system is consisted of a central control unit that handles information gathering from multiple mobile sources and a set of mobile agents capable of domain data monitoring and measurement reporting.

The Control unit (CU) plays a central role in registration of mobile agents. It is responsible for management of the measuring and reporting. The CU has to store data for mobile agents and operation-related data of the telemetry system. The CU needs to provide interfaces to a measurement data repository. The mobile agent (MA) belongs to the class of embedded devices, equipped with sensor(s), positioning module, data transmission module and power supply module. The main requirement to the MA is to operate using as low energy as possible.

The access to measurements database may be provided by Web Services. Web Services is pervasive technology that enables building of network of services. The technology allows flexible implementation of more sophisticated applications by combining the existing Web Services with the Parlay X ones.

If Web Services are used for open access to measurements data, the owner of the measurements repository has to deploy a gateway that exposes APIs toward 3rd party applications and Database Management System (DBMS) operations toward the measurements repository as shown in Fig.1. Internal Web Services API may be used by the CU to supply data to the gateway.

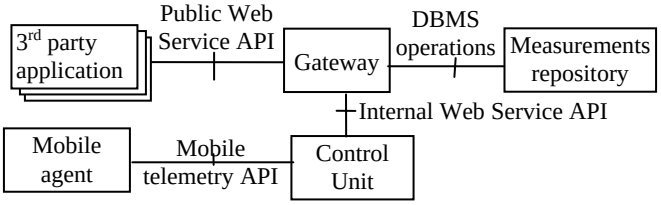

Fig.2 Functional entities involved in open access to mobile telemetry data

Common functions for mobile telemetry applications include the following: MA registration, communication security, operational mode management, CU notifications to MA, measurements reports, and error handling. The MA may be configured to operate in different operation modes. Periodic monitoring and reporting requires the MA to make measurement and send reports periodically. Triggered reporting operation mode requires monitoring for certain criteria and submitting reports on their occurrence. The third mode of operation is reports upon request. The CU induced reporting requires MA to perform measurements and report on demand.

III. MOBILE TELEMETRY API

The proposed approach to design APIs for mobile telemetry defines two interface packages: one supported by the mobile agent and another one supported by the central unit. The generic interface **MobileAgent** is inherited by interfaces which provide functions for the MA client (**MaClient**) and interfaces which provide functions for the MA server (**MaServer**). The interfaces of the MA client have prefix 'MaC', while the interfaces of the MA server have prefix 'MaS'. The generic interface CentralUnit is inherited by interfaces which provide functions for the CU client (**CuClient**) and interfaces which provide functions for the CU server (**CuServer**). The interfaces of the CU client are named by prefix 'CuC', while the interfaces of the CU server are named by prefix 'CuS'. Fig.2 shows the structure of APIs for mobile telemetry.

The MA supports the following interfaces. The **MaSPriodicReport** interface is used by the CU to configure the MA operational mode for period measurements reporting. The interface provides methods for initiation, modification and termination of periodic reporting. The **MaSTriggered-Report** interface is used to configure the MA operational mode for triggered measurements reporting. The interface provides methods for initiation, modification and termination of reporting in case of occurrence of specific events. The **MaSOnDemandReport** interface is used by the CU to request a measurements report ad hoc. The **MaCMea-surementsReport** callback interface is used to receive results of measurements reporting. The **MaCRegistration** is callback interface used to receive results of the registration request. The **MaSAdministration** interface is used for administrative purposes. It provides methods which allow CU to request authentication, to send new temporary identities and to send notifications to the MA.

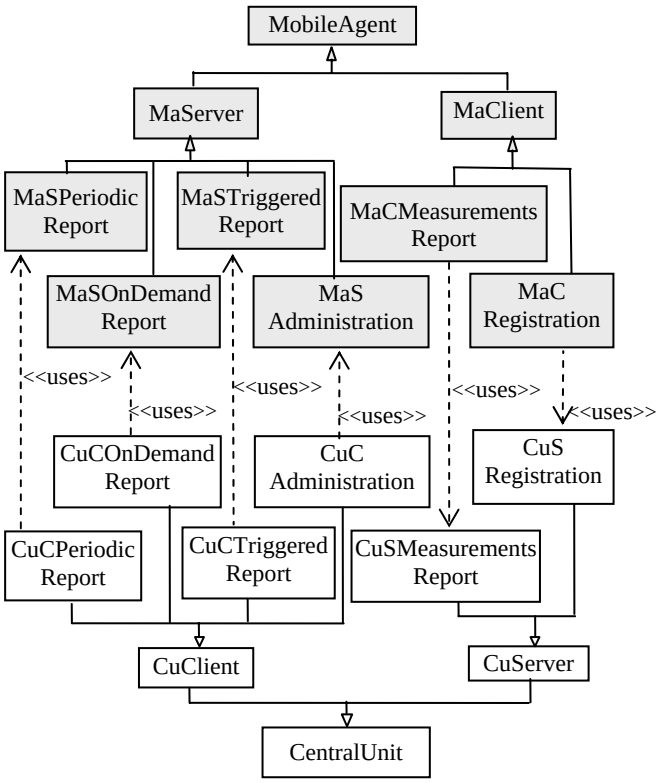

Fig.2 Mobile telemetry interface structure

The CU supports the following interfaces. The **CuCPeriodicReport, CuCTriggeredReport** and **CuCOnDemanReport** are callback interfaces. They provide methods which may be used to receive results of operation mode configuration requests. The **CuSMeasurementsReport** interface is used by the MA to report measurements periodically, triggered and on demand. The **CuSRegistration** interface is used by the MA to maintain its registration. The **CuCAdministration** is callback interface which is used to receive the results of the authentication requests, of new temporary identities allocation requests and of notification requests.

IV. Web Service Access to Measurements

Two Web Services may be defined to maintain consistent information flows between the CU and the gateway and between the gateway and 3rd party applications.

The **Monitoring Management** Web Service allows the CU to supply data to the gateway. It is internal for the mobile telemetry system and is not for public usage. The **Monitoring Management** interfaces are divided into two categories: implemented by the gateway and implemented by the CU. The interfaces implemented by the gateway are **MeasurementsManagement** interface, **Authentication** interface, and **CUNotificationManager** interface. The CU implements the **CUNotification** interface.

The **MeasurementsManagement** interface supports the following operations:

- **submitMeasurements()** operation is used by CU to submit measurements to the database;
- **modifyMeasurements()** operation is used to update previously submitted measurements data;
- **deleteMeasurements()** operation is used to delete previously stored measurements data;
- **queryMeasurements()** operation is used to query for measurements data. The query may include expressions that define the selection conditions.

To make the mobile telemetry system scalable the security related parameters may be stored at AAA server (not shown in Fig.1). The **Authentication** interface supports the following operation:

- **getAuthenticationData()** operation is used to request the authentication parameters related to given MA and stored at the AAA server. The parameters are required by CU to allow building the requests for authentication to the MAs.
- **updateAuthenticationData()** operation is used to store the new temporary identities of the CU and the MA in case of successful authentication.

The **CUNotificationManager** interface supports the following operations:

- **startCUNotification()** operation sets up the notifications in order to inform an authorized user through the administrative interfaces of the CU that measurements which met the defined criteria occurred. When such measurements are submitted, the user is notified about corresponding criteria.
- **endCUNotification()** operation terminates the previously installed triggers for notifications based on given set of criteria.

The **CUNotification** interface is implemented by the CU and supports the following operation:

- **notifyAlertingMeasurements()** operation allows to receive notifications when submitted measurements reach previously defined thresholds.

The **Measurements Access** Web Service is part of the public Web Service which allows 3rd party applications to access measurements data. The Web Service also provides subscription/notification interfaces which allow a 3rd party application to get a notification when particular thresholds in measurements data are reached.

The **DataAccess** interface is implemented by the gateway and it supports the following operation:

- **readMeasurements()** operation allows an authorized 3rd party application to read stored measurements data. The operation specifies the selection conditions.

The **AppNotificationManager** interface is implemented by the gateway and it supports the following operations:

- **startAppNotification()** operation is used by a 3rd party application to register its interest in receiving notifications about specific events in the measurements dataset.
- **endAppNotification()** operation is used by the 3rd party application to indicate that it is no longer interested in receiving notifications.

The **AppNotification** interface has to be implemented by the 3rd party application and supports the following operation:

- **notifyAppMeasurements()** operation allows to receive a notification after specific events defined by **startAppNotification()** operation occurred in the dataset.

The Web Services technology does not provide standardized means for authentication and authorization. Framework interfaces are defined as a part of **Measurements Access** Web Service. These interfaces provide authentication functionality for external access to measurements database. The 3rd party application authentication and authorization parameters may be stored at the AAA server.

The **GWFramework** interface is implemented by the gateway and it supports the following operation:

- **gwAuthentication()** operation is used by a 3rd party application to authenticate the gateway.

The **AppFramework** interface is implemented by the 3rd party application and it supports the following operation:

- **appAuthentication()** operation is used by the gateway to authenticate the 3rd party application.

The **ServiceAgreementManagement** interface is implemented by the 3rd party application and it supports the following operations:

- **signServiceAgreement()** operation is used by the gateway to request the 3rd party application to sign an agreement on the service;
- **terminateServiceAgreement()** operation is used by the gateway to terminate an agreement for the service.

Fig.3 shows the message sequence when a 3rd party application authenticates with the gateway.

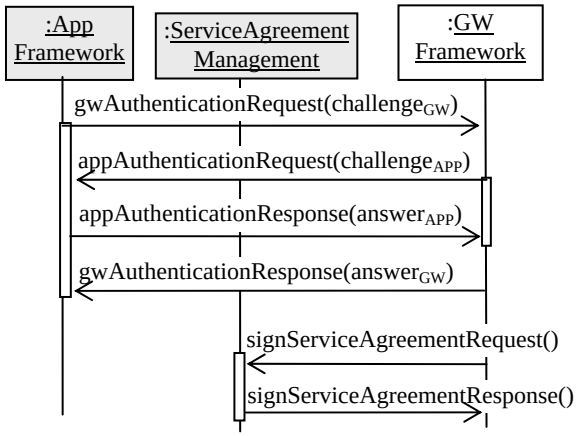

Fig.3 Third party application authentication and service agreement

V. USE CASE

An example application that provides ubiquitous access to local CO_2 emissions levels may use the service brokering function provided by the gateway and the Parlay X **Terminal Location** Web Service defined in 3GPP TS 29.199-9.

Parlay X Web Services are intended to provide simple interfaces for open access to functions in the public communication network. The Parlay X **Terminal Location** Web Service provides access to the positioning information through request or notification of a change in the location of a terminal or notification of terminal location on a periodic basis. The terminal location is expressed through a latitude, longitude, altitude and accuracy.

Fig.4 shows the sequence diagram for a hypothetical application "Can I breathe here?"

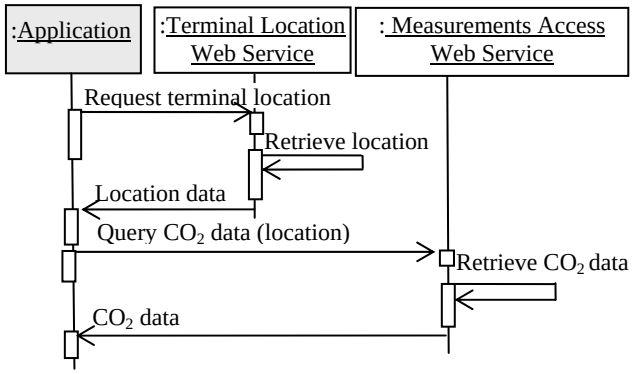

Fig.4 Hypothetical application "Can I breathe here?"

The "Can I breathe here?" application displays at the mobile subscriber terminal a map with subscriber's position and local CO_2 emission levels. It uses the Parlay X **Terminal Location** interfaces to query about terminal location and **Measurements Access** Web Service interfaces to access the CO_2 measurements data. For the application to determine the location of a mobile device, it provides a mobile subscriber's terminal address and desired accuracy, and then it receives the location for the terminal requested. When querying the CO_2 measurements database, the application provides the desired location of the measurements and receives information about current CO_2 levels at its location.

VI. CONCLUSION

The paper provides a new structural approach to definition of Application Programming Interfaces for mobile telemetry. Derived from generic functions the APIs provide an abstraction of mobile telemetry that is independent of the protocols used for implementation. The open access to measurements data allows 3rd party applications to access data in a secured manner. The proposed Web Services for open access to measurements data accelerate the development of new attractive applications and shorten the time to market. The suggested solution for internal access to measurements repository though Web Services interfaces provides an abstraction of DBMS operations which is an additional level of flexibility i.e. the changes of the storage technology will not impact on the overall function of the system. The future work will be focused on API design, implementation and test.

ACKNOWLEDGEMENT

The research is in the frame of Project DDBY02/13/2010 funded by NSF, Ministry of Education Youth Science, BG.

REFERENCES

[1] M. Alenazi, S. Gogi, D .Zang, E. Cetinkaya, J. Rohrer and J.Sterbenz. "ANTP Protocol Suite Software Implementation Architecture in Python", International Telemetry Conference ITC'2011, Conference Proceedings, pp.1-10, 2011.

[2] D. Forthoffer, M. Throop and B. Tilman. "Mobile Telemetry System", Patent Application Publication, Pub. No.US 2011/-166741 A1, 2011.

[3] B. Chi, J. Yao, S. Han, X. Xie, G. Li, and Z. Wang. "Low-Power Transceiver Analog Front-End Circuits for Bidirectional High Data Rate Wireless Telemetry in Medical Endoscopy Applications", IEEE Transactions on Biomedical Engineering, vol. 54, issue 7, pp. 1291–1299, 2007.

[4] B. Townsend, J. Abawajy, and T Kim. "SMS-Based Medical Diagnostic Telemetry Data Transmission Protocol for Medical Sensors", Journal on Sensors, 2011.

[5] M. Cibuk, and H. Balik. "A novel solution approach and protocol design for bio-telemetry applications", Advances in Engineering Software, vol. 42, issue 7, pp. 513-528, 2011.

[6] S. Ambler. "Encapsulating Database Access: An Agile "Best" Practice", http://www.agiledata.org/essays/implementation Strategies.html.

[7] D. Jayasinghe. "Exposing a Database as a Web Service", http://www.developer.com/db/article.php/3735771/ Exposing-a-Database-as-a-Web-Service.htm, 2001.

[8] J. Yang, Z. Zhang, and Y. Zhao. "Analysis on Database Connection Mechanism of WEB Application System in Dreamweaver", ICITA'2010 Wuhan, China, 2010, Conference Proceedings pp. 1-4, 2010.

[9] X. Qu, J. Feng, and W. Sun. "United access of distributed biological information database based on web service and multi-agent", Control and Decision Conference, Yantai, Shandong, China, Conference Proceedings, 2008.

[10] T. Sellis, D. Skoutas, and K. Staikos. "Database interoperability through Web Services and ontologies", 8th IEEE International Conference on BioInformatics and BioEngineering, 2008, Conference Proceedings pp.1-5.