An Approach to Design Web Services for Remote Entity Management

Ivaylo Atanasov[1], Nikolay Krastanov[1] and Evelina Pencheva[1]

Abstract – **In this paper an approach to design web services for device management in Machine-to-Machine applications is studied. Based on the requirements analysis, functional abstraction for fault and performance management, and configuration management is synthesized. Typical use case scenarios are modeled and web service interfaces are described.**

Keywords – **Machine-to-Machine communications, Service Oriented Architecture, fault and performance management, configuration management**

I. INTRODUCTION

Machine-to-Machine (M2M) communications connecting smart objects using Internet Protocols proved to be the next successful step in Internet evolution. With the growth of M2M devices, it is expected that the M2M communications will change the network traffic profile in near future [1].

Remote entity management encompasses complex operations that are quite orthogonal to the M2M application business logic [2]. ETSI defines M2M service capability (SC) as a function that is to be shared by different applications. Remote entity management (xREM) M2M SC provides functions pertaining to general management, e.g. configuration management, diagnostics and monitoring management, software/firmware management, area network management and SC layer management [3].

While the research is focused on software/firmware management [4], less attention is paid on the diagnostics and monitoring management and configuration management due to their specifics. Configuration management allows configuration of the device capabilities and features for supporting M2M services and applications. Diagnostics and monitoring management allows running specific diagnostics tests on a device and collecting the results or alerts, and this package is also called fault and performance management.

Most of the existing solutions for device management [5], [6] use advanced technologies such as TR 069 of the Broadband Forum and OMA Device Management (DM). Both protocols feature own specifics. For example, OMA DM is a protocol for management of mobile devices such as mobile phones and tablets that provide functions for software configuration and update, fault and performance management [7], [8]. The protocol requires session establishment for device management by sending a special short message to the device. TR 069 CWMP (CPE WAN Management Protocol) is a management protocol used in a cable environment and does not support short messaging, hence, forcing the device to create session management requires other mechanisms [9], [10]. CWMP is a protocol for remote management of end-user devices and as a bidirectional SOAP/HTTP based protocol, it provides the communication between customer premises equipment and auto configuration servers. It is also used as a protocol for remote management of home network devices and terminals, and there is a growing trend for the use in M2M communications.

Because of its expressiveness, the technology of Web services is very suitable for M2M communications [11]. There are several ways to realize the concept of Web services. Some solutions are built on mechanisms, which require significant processing power and communication resources, while others are more lightweight. Devices with constrained resources need to use more simple mechanisms. REST (Representational State Transfer) is an architectural style that represents the simplified Web services. REST is a set of principles that provide greater extensibility of distributed systems and allow an increase in the number and variation of distributed applications due to the weak dependence of the components. The research focus in this paper is on REST-based web services for remote entity management. We present an approach to design web services for remote entity management.

The paper is structured as follows. Section II provides a mapping of Mobile telemetry functional architecture presented in [12] onto ETSI M2M System Architecture. In Section III, a functional abstraction of fault and performance management and configuration management is synthesized by identification of use case scenarios, interface class diagrams and state diagrams are described. The conclusion summarizes the authors' contributions.

II. MAPPING OF TELEMETRY FUNCTIONAL ARCHITECTURE ONTO ETSI M2M SYSTEM ARCHITECTURE

The functional architecture of the Mobile telemetry system and the supported Service platform for ubiquitous access to measurements, presented in [12], may be mapped onto ETSI M2M System Architecture [3], as shown in Fig.1. The system consists of the following functional entities.

The Mobile Agent (MA) is a smart object and it is responsible for the spatio-temporal measurements in the application area. The Control Unit (CU) is responsible for the control over multiple distributed MAs, and for the transmission of measurements provided by MAs to a centralized database through a proxy function. The Database server stores measurements data. The Proxy function provides access to the measurements database for authorized CUs when

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

new measurements data have to be recorded. The AAA server provides authentication, authorization and accounting functions for MAs, CUs and 3rd party applications that try to access information about measurements data. The Measurements gateway provides the functions for secured service brokering function for 3rd party applications to access the measurements data. The Management Centre provides the following management functions: real-time access for administrative purposes; management of the Mobile telemetry system; user interface is required to configure the MA operation mode, to handle data in the AAA server, to manage the processes in Measurements gateway and Control units. The Application server hosts external applications that use accumulated measurements in the application domain. All the system entities share information to accomplish the common objective related to monitoring and ubiquitous access to measurements. The entities communicate through interfaces that are defined by Application Programming Interfaces – APIs or communication protocols.

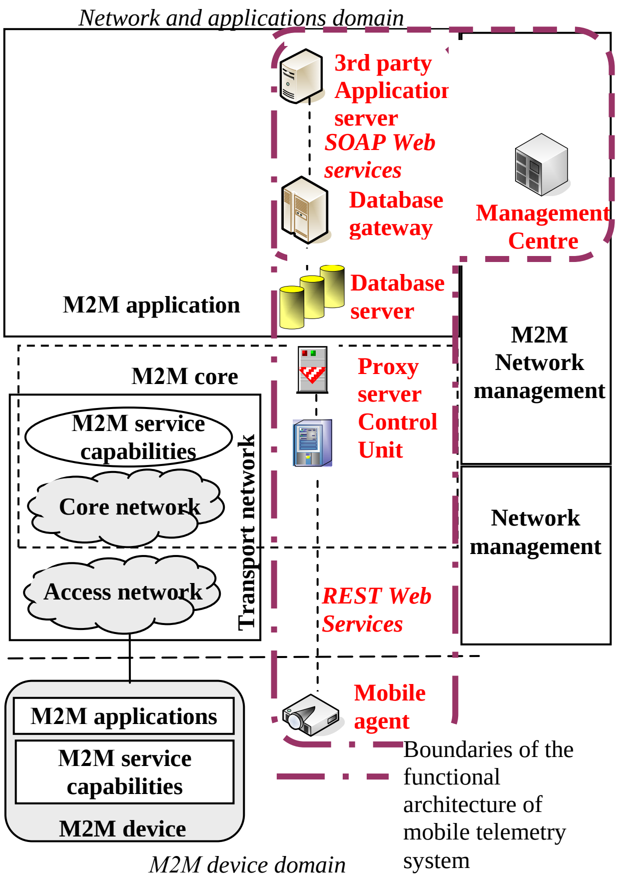

Fig. 1. Mobile telemetry functional architecture mapped onto ETSI M2M System Architecture

The ETSI high level M2M system architecture provides end-to-end M2M system description, in which not all functions are standardized [3]. The ETSI M2M device performs M2M applications using M2M service capabilities and network functions. This element corresponds to the MA. MA is equipped with a wireless communication module, and receives direct access to the access network operator. The MA can be connected directly to the infrastructure of the 2G, 3G or 4G. Mobile agents perform procedures such as registration, authentication, authorization, management and configuration of network and applications domain.

In the ETSI network and applications domain, M2M service capabilities are available for network applications and the operator core network. Core network provides connectivity functions, network services and management functions, and the AAA server is part of the core network. M2M service capabilities provide features that are available to network applications in a variety of open interfaces. M2M applications perform service logic and use M2M service capabilities through open interfaces. In the Mobile telemetry system, specific M2M applications are running in the Control unit, Proxy server, Database server, Database gateway and Application server. M2M management includes all functions necessary for management of M2M applications and M2M service capabilities. In the Mobile telemetry system, M2M management functions are performed by the Management center. More detail on ETSI M2M System Architecture may be found in [3].

In the Mobile telemetry system, the Database gateway interfaces may be implemented as SOAP-based Web services [13], while interfaces between the Control Unit and Mobile agent may be realized as REST-based Web services. The advantages of the proposed functional architecture of Mobile telemetry system in comparison with related works [14] [15] is not a subject of the paper and may be found in [12].

III. Functional Abstraction for Remote Entity Management

The aim of fault and performance management is to take corrective measures to ensure service continuity. Fault and performance management applies to detecting faults pertaining to all software and hardware components of the device, as well as monitoring performance indicators. Configuration management applies to setting up different parameters of the device to allow its proper operation.

Web Services for device management need to provide a high level of functional abstraction by defining a set of interfaces with operations and information abstraction by defining data structures.

We define Integrity management interfaces to provide functions ensuring that both the device M2M application and network M2M application will not be overloaded by making excessive number of requests.

Fig.2 shows a sequence diagram for the scenario of subscription, suspending and resuming notifications from device M2M application based on evaluation of local policy as a result of detection of load level change of the device M2M application.

The network M2M application uses startLoadLevelNotifications(), suspendLoadLevelNotifications(), resumeLoadLevelNotifications(), and stopLoadLevelNotifications() operations to manage its subscription to notifications about changes in device M2M application load level. The reportLoadLevel() operation is invoked by the device M2M application when a load change condition is detected, e.g. three load levels may be defined – normal (not congested or overloaded), overloaded, and severely overloaded. The application uses this operation to report its current load level.

The sequence diagram in Fig.3 shows how the network M2M application requests load statistics from a device M2M application. The queryLoadStatistics() operation is used to request the application to provide load statistics records.

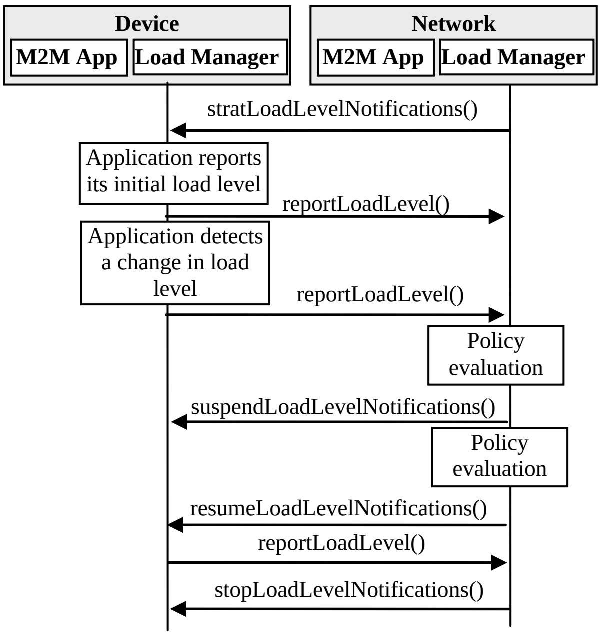

Fig. 2. Load Management: start/suspend/resume/stop notifications from device M2M application

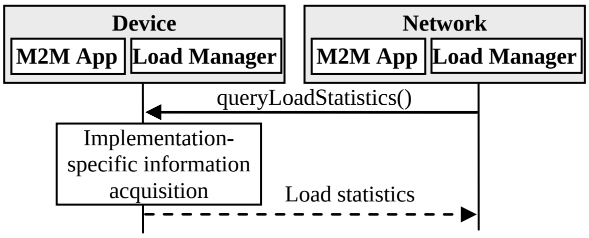

Fig. 3. Load Management: network application queries load statistics

We define Availability management interfaces to exchange information that affects the integrity of the device M2M application and network M2M application.

Fig.4 shows the scenario where the network M2M application decides to monitor the device M2M application, and therefore requests the device M2M application to start reporting periodically its availability status (startAvailStatus-Reporting() operation). The device M2M application responds by reporting its availability status at specified intervals (reportStatus() operation). The network M2M application decides that it is satisfied with the device M2M application availability status and stops availability status reporting (stopAvailStatusReporting() operation).

We define Fault management interfaces to exchange information about operational M2M application status.

Fig.5 shows the scenario where the network M2M application has detected that the device M2M application has failed (probably by the use of availability reporting mechanism). The network M2M application informs the device Fault Manager by invoking the availStatusInd() operation.

It is also possible for device M2M application to ask the network M2M application to do activity test by invoking activityTest() operation, e.g. due to the lack of respoces. The network M2M application does the activity test and sends the result to the network M2M application as shown in Fig.6.

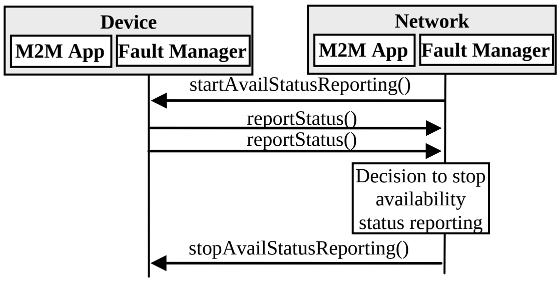

Fig. 4. Availability Status Management: start/report/stop availability status supervision of the device M2M application

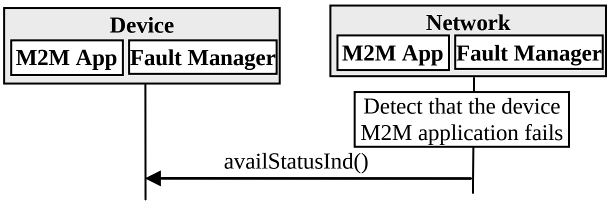

Fig. 5. Fault Management: indication that the device M2M application has failed

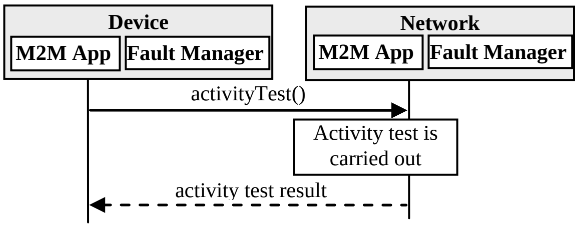

Fig. 6. Fault Management: device M2M application requests network M2M application activity test

Fig.7 shows the interface class diagram of interfaces supported by the device M2M application. Fig.8 shows the interface class diagram of interfaces supported by the network M2M application.

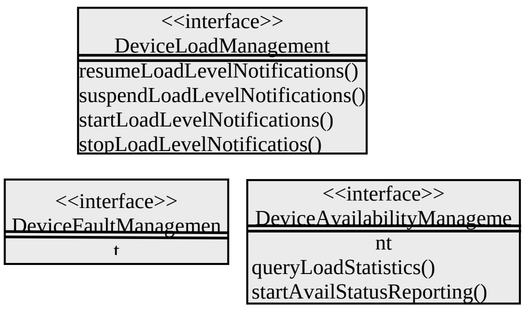

Fig. 7. Interfaces supported by device M2M application

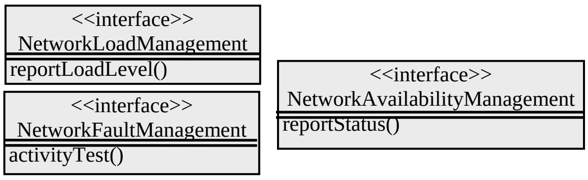

Fig. 8. Interfaces supported by network M2M application

In REST-based architecture, communications are stateless, and each request can be handled independently from the others. However, a management application needs to maintain states of the remote entity in order to execute the processing logic. Fig. 9 shows the state transition diagram of the Device Load Manager. In Active state, load statistics information may be obtained. In Notification Suspended state, the load reporting is suspended due to e.g. a temporary load condition.

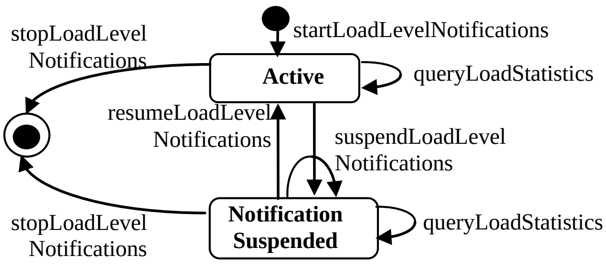

Fig. 9. State transition diagram of the Device Load Manager

Fig. 10 shows the state transition diagram for fault management. Active state is the normal state, which is fully functional and able to handle requests. In Faulty state, an internal problem is detected.

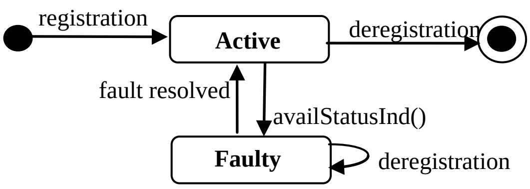

Fig. 10. State transition diagram of the Device Fault Manager

Among the others, configuration management includes configuration of device mode of operation. For mobile telemetry applications, the Mobile Agent may be configured to perform and report periodic measurements, triggered measurements and on demand measurements. Fig. 11 shows the class diagram of configuration management interfaces.

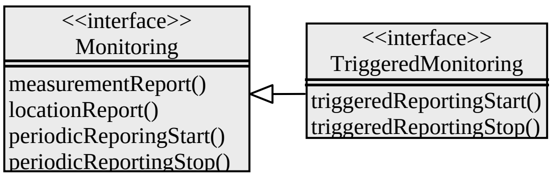

Fig. 11. Interfaces for device's operation mode configuration

IV. Conclusion

The paper presents functional mapping of Mobile telemetry system architecture onto ETSI M2M System architecture.

The usage of Web services for Mobile Telemetry System is discussed. An approach to design web services for remote entity management in M2M communications is studied. The approach is based on synthesis of functional abstraction for fault, performance and configuration management. Functions for load management, availability status management, fault management, and operation mode configuration are identified. Sequence diagrams are used to model typical use case scenarios. Respectively, class diagrams describe the Web services interfaces and operations, and state diagrams model the management application's view on the remote entity state that is managed.

By using Web service for M2M applications, existing web-service oriented business systems and knowledge can be applied to the growing field of smart object communications.

Acknowledgement

The work is under the support of the research project №141ПР004-7 funded by RDS, TU-Sofia.

References

[1] T. Potsch, S. Marwat, Y. Zaki, C. Gorg, "Influence of future M2M communication on the LTE system", Proc. of Wireless and Mobile Networking Conference (WMNC), 2013, pp.1-4.
[2] H. Zhou, The Internet of Things in the Cloud, A Middleware Perspective, CRC Press, 2012.
[3] ETSI TS 102 690 "Machine-to-Machine (M2M) Functional architecture, v.1.1.1, 2010.
[4] Y.C. Chang., T.Y.Chi, W.C. Wang, S.Y. Kuo, "Dynamic software update model for remote entity management of machine-to-machine service capability", IET Communications, vol.7, issue 1, 2013, pp.32-39.
[5] J.-P. Vasseur, A. Dunkels, "Interconnecting Smart Objects with IP, The next Internet", Morgan Kaufmann, Elsevier, 2010.
[6] L. Barrig, B. Smeets, K. Sallberg, "M2M Remote-Subscription Management", Ericsson Review, 2011.
[7] M. Unmehopa "Management and Provisioning of M2M Devices for a Connected World", ETSI M2M Workshop, 2011.
[8] OMA-AD-DM-V1_3 Device Management Architecture, version 1.3, 2010.
[9] Broadband Forum, "TR-069 CPE WAN Management Protocol", version 1.4, Technical report, 2013.
[10] J. Bernstein, T. Spets, G. Bathrick, G. Pitsoulakis, DSL Forum "TR-069 CPE WAN Management Protocol", 2004.
[11] P. Balamuralidhara, P. Misra, A. Pal, "Software Platforms for Internet of Things and M2M", Journal of the Indian Institute of Science. Vo.93, No.3, 2013.
[12] I. Atanasov, E. Pencheva. "Towards a Scalable Mobile Telemetry System and Ubiquitous Access to Measurements", International Journal of Reasoning-based Intelligent Systems (IJRIS), 2013, vol.5, no.1, pp 51-62.
[13] I. Atanasov, V. Trifonov, K. Nikolova, E. Pencheva. "Database Abstraction Layer for Access to Mobile Monitoring Measurements", International Journal on Information Technology and Security, 2012, vol. 4, pp.13-30.
[14] 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.
[15] D. Forthoffer, M. Throop, and B. Tilman. "Mobile Telemetry System", Patent Application Publication, Pub. No.US 2011/-166741 A1, 2011.