Message based integration in Cyber-Physical System: firefighters in the field

Tiago Magalhães, Ilídio Castro Oliveira, and José Maria Fernandes
IEETA, Dept. of Electronics, Telecommunications and Informatics (DETI)
University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
{tiagomagalhaes, ico, jfernan}@ua.pt

ABSTRACT

We present a Cyber-Physical System for distributed monitoring of first response teams. To cope with different setups, different sensors and computing devices, found in the evolution of the VitalResponder project, we adopted a message brokering pattern to isolate the data aggregation logic from the changing hardware and networking configurations.

Sensors publish data to local message brokers (body-area level), that, in turn, publish to higher level messages queues (site-wide level). The brokers use adapters to relay messages over alternative transport protocols, currently available as adapters for messaging standards over TCP (AMQP, MQTT), over UDP (CoAP), and basic sockets (TCP and UDP).

The abstraction of messages brokering facilitates the portability to different platforms, and the system has been tested in Android, Raspberry Pi and standard Linux devices, to support a distributed sensor network for firefighter's missions monitoring.

Categories and Subject Descriptors

Embedded and cyber-physical systems; Ubiquitous and mobile computing systems and tools; Publish-subscribe/event-based architectures; Life and Medical Sciences

General Terms

Design, Experimentation, Standardization.

Keywords

Message brokering, Cyber-Physical Systems, monitoring, firefighters, Android OS, Raspberry Pi, CoAP, AMQP.

1 INTRODUCTION

Under the VitalResponder project[1] we have developed solutions to monitor firefighters during forest fires events. Our initial concerns were on monitoring firefighter physiological status and location [1, 2] and, later, to ensure connectivity through wireless network transport [3]. In these scenarios we have monitored firefighters' ECG using wearable sensors[2], their location based on GPS (phone or external) and their involving environment using in-house solutions (CO and NO_2 levels, temperature, atmospheric pressure, altitude, humidity, and luminosity [3]).

In this work we describe our Cyber-Physical System (CPS) for monitoring firefighters during forest fires events. We focus the message-oriented architectural style that was implemented to build a highly decoupled distributed monitoring system.

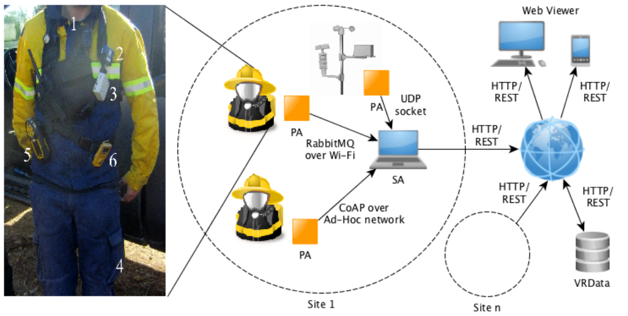

Figure 1: System architecture composed by several sites each one containing Personal Aggregators (PA) and a Site Aggregator (SA). Vital and environmental monitoring sensors: 1) VitalJacket, 2) FREMU, 3) Helmet Unit, 4) external GPS and smartphone, 5) GasAlertMicro 5 PID, and 6) GasAlert CO.

2 ARCHITECTURE

2.1 Message queues for distributed monitoring

CPS nodes can be either Personal Aggregator (PA) or Site Aggregator (SA). The first are local/body-area aggregators of sensors data, namely wearable or other Bluetooth connected sources. The SA extends PA acting as a gatherer for multiple PA information and as data gateway/provider to external clients.

Our system uses messaging to implement a CPS message data bus in which nodes 1) submit message with a given topic/queue or 2) receive/check messages from a given topic/queue. The message brokering logic is used both at the local aggregator and across the network:

[1] `http://vitalresponder.web.ua.pt/`

[2] `http://www.vitaljacket.com/`

MOBIQUITOUS 2015, July 22-24, Coimbra, Portugal

DOI 10.4108/eai.22-7-2015.2260246

Table 1: Local brokering implementation.

Implementation	Local brokering solution
Android	Brokering is mapped into the Android Intents mechanisms.
Python	RabbitMQ

Table 2: Broker and network resources mapping.

Implementation	Network brokers solution
TCP	AMQP MQTT
UDP	CoAP, adapted with tags.
TCP/UDP sockets	Sockets with extra tags.

Local level: CPS nodes are responsible for messaging handling containing a broker that can be locally accessed. Sensor reading acquisition processes can access the CPS by using predefined message topics that are used as placeholders of information that can be later forward according to the topic semantic and the configured network transport layer. The messages arriving to the local broker are relayed to the proper network driver that sends it to the CPS.

Site level: SA performs the role of logical message broker for the CPS nodes and external consumers, in which topics are used to capture domain specific data pathways and also to allow explicit exchange with public resources in the CPS (e.g. VRData repository or Web Viewer - Figure 1).

2.2 Extensions to support the Logical Message bus abstraction

CPS wide messages are sent/received through topics (e.g.: vr2.sensors.gps, vr2.sensors.vj, vr2.sensors.helmet). This allows an easy integration of new sensors or data source with the CPS as the only thing required for integrating is a known IP address to a PA or SA, connectivity, and use the predefined "topics" to publish or retrieve information.

The logical mechanism of messaging broker is easily implemented in different platforms, given that there are several efficient middlewares available.

At the local aggregator, the PA broker provides a device public messaging resource that can be used by processes running in the device. It has been implemented in Android and in Python, to address Raspberry Pi devices (Table 1).

At the site level, the broker has been mapped into a messaging middleware (Table 2).

Sockets and CoAP imply appending to the message a domain specific tag, encoding the topic/queue information used to allow the receiving SA to unpack and relay incoming message to the SA broker, as they do not support these concepts.

From the example 1, in python, the input is read from the sensor via Bluetooth. This input (in this case ASCII text) is placed in the message and published in the PA internal broker intent channel (gateway data thru channel "vr2_intents" and intent "pt.ua.ieeta.vr2.fremu.values") to be, later, exported to the site-wide data bus as "vr2.sensors.fremu" (i.e. routing key for values from sensor FREMU – an environment sensor unit).

The PA broker directs the local incoming messages from "vr2_intents" channel to the SA broker using the configured network resource (Table 2).

Example 1: Python excerpt to connect sensor to CPS.

```
# Receive data from the sensor and package
    the message
message = self.read(sensorsocket)
message = self.parse(message)
# Publish locally via RabbitMQ
channel.basicpublish(exchange="vr2_intents"
    , routingkey="pt.ua.ieeta.vr2.fremu.
    values", body=message)
```

3 CONCLUSION

In this paper we describe a CPS for firefighters monitoring, supported on messaging based approach that provides a working and tested solution while abstracting the CPS nodes from the heterogeneous nature of the node and the network transport topology. The current implementation of the CPS was successfully tested in laboratory and field, settings supported on both Android and Raspberry PI based nodes using TCP and UDP either as sockets or over standard messaging protocols, such as AMQP, MQTT, and CoAP (using Californium). Although our system target is a very specific scenario with specific sensors and requirements, we are evolving it to allow configuring the CPS specific topics to a more agnostic CPS semantics.

4 ACKNOWLEGMENTS

This work was supported by the European Regional Development Fund (FEDER) through the COMPETE programme and by the Portuguese Government through the FCT, Fundação para a Ciência e a Tecnologia, in the scope of the projects Incentivo/EEI/UI0127/2014 and UID/CEC/00127/2013, (IEETA/UA, www.ieeta.pt), PTDC/EEI-ELC/2760/2012 (VitalResponder 2.0) and CMUP-ERI/FIA/0031/2013 (VR2Market, by CMU-Portugal program).

5 REFERENCES

[1] M. Colunas, J. Fernandes, I. Oliveira, and J. Cunha, "Droid jacket: Using an android based smartphone for team monitoring," in *IWCMC, 2011 7th International*, July 2011, pp. 2157–2161. [Online]. Available: http://dx.doi.org/10.1109/IWCMC.2011.5982868

[2] D. Teles, M. Colunas, J. Fernandes, I. Oliveira, and J. Cunha, "ivital: A real time monitoring system for first response teams," in *Mobile Networks and Management*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, K. Pentikousis, R. Aguiar, S. Sargento, and R. Agüero, Eds. Springer Berlin Heidelberg, 2012, vol. 97, pp. 396–404. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30422-4_29

[3] A. Aguiar, E. Soares, P. Brandão, T. Magalhães, J. M. Fernandes, and I. Oliveira, "Demo: Wireless ip mesh on android for fire-fighters monitoring," in *Proceedings of the 9th ACM MobiCom Workshop on Challenged Networks*, ser. CHANTS '14. New York, NY, USA: ACM, 2014, pp. 89–92. [Online]. Available: http://doi.acm.org/10.1145/2645672.2645692