Demo Abstract: Monitoring Co-Movement of Smart Objects using Accelerometer Data

Luis E. Talavera, Markus Endler
Department of Informatics
Pontifícia Universidade Católica do Rio de Janeiro
{ltalavera, endler}@inf.puc-rio.br

Francisco Silva da Silva
Department of Informatics
Federal University of Maranhão
fssilva@deinf.ufma.br

Abstract—In the Internet of Mobile Things (IoMT) applications dealing with mobile objects may benefit from information if two or more objects are in co-movement, i.e. if they are close to each other and sharing the same movement pattern during some interval of time. In this demo we demonstrate how co-movement can be reliably monitored in a mobile middleware component executing on a smartphone using Complex Event Processing over streams of accelerometer data generated by SensorTags interacting with the smartphone through Bluetooth Low Energy.

Index Terms—Internet of Mobile Things, Mobile Sensor Analysis, Middleware, Co-movement

I. INTRODUCTION

So far, very few research has focused on the Internet of Mobile Things (IoMT), in which the Things (or Objects) can be moved - or can move autonomously - and yet remain connected for remote access or control from anywhere in the Internet despite having only low-range wireless communication capabilities, such as NFC, Bluetooth or ANT+. Mobile Objects in IoMT can be of and type and complexity, ranging from sophisticated mobile robots, all sorts of vehicles and moving machines, personal gadgets, wearable devices, down to simple sensors, tags or badges, all of them with their very specific wireless capabilities. This differs from the majority of approaches towards IoT, where the peripheral devices, the smart objects, are assumed to be either: stationary and always connected to some propagator node with Internet connectivity, or else, have their own Internet connectivity and are location aware. However, when we consider most of today’s wearables, gadgets or smart objects, only a minimal fraction of them have embedded Internet connectivity and location awareness. Thus, to provide Internet connectivity to all these peripheral devices, there is nothing less suitable than the most ubiquitous, affordable - and mobile - propagator device: the end-user mobile devices (smart phones and tablets). Hence, we advocate the use of a generic middleware entity, the Mobile Hub (M-Hub) as an enabler of the IoMT paradigm (see section II).

Another fundamental challenge in IoMT centers around processing of sensor data, typically produced in a very dynamic environment, which is not only extremely large in volume and noisy, but also continuous in form of data streams. In the M-Hub concept we also address this issue by providing functionalities for performing filtering, aggregation and preprocessing of the data originated from the smart mobile objects. In particular, it can also detect high-level events that represent some relevant pattern over the collected data streams using a Complex Event Processing engine. This approach provides flexibility and scalability, since the processing power required for monitoring/tracking a large amount of mobile objects can be divided between the M-Hubs and a cloud infrastructure.

Defining Co-Movement

In IoMT with general and unrestricted mobility of propagator devices and smart objects, beyond simple tracking or basic queries such as "where is object X now?", applications dealing with mobile objects may require more sophisticated information, e.g., if the object(s) of interest is/are being transported - or carried by - another trackable object, such as a vehicle. In particular, Co-movement, appears as a fundamental relation among objects in IoMT applications, since it may be reveal information about the current state of transportation, of goods, machines and persons. We define that a set of objects are in mutual co-movement state at time $t$ iff they have stayed within a given distance interval $D$ from each other, and shared very similar speeds and accelerations during the time interval $[t−δ, t]$. However, since continuous verification of co-movement is impossible, we should re-phrase the part of the definition regarding the time interval to ... for the $N$ most recent data probes.

With information about the kind of objects involved (say X and Y), the knowledge of co-movement may reveal important information about them, such as: X is being carried by Y, X and Y are on the same path or X and Y should arrive at the same destination, etc. And such information may be quite relevant for IoT applications in industry, logistics, personal and public security, m-health, tourism, etc. For example, co-movement could reveal: a passenger (i.e. its smartphone) is within a connected smart car or train/bus, smart parcel items are on a same smart truck or vessel, terrestrial or aerial robots are in a swarm, or else, a health sensor is carried by a user with a smartphone, etc.

In this demo we demonstrate how co-movement can be reliably monitored for Sensor Tags\footnote{Texas Instruments CC2541 Sensor Tag - http://www.ti.com/lit/ml/swru324b/swru324b.pdf} that are equipped with...
Bluetooth Low Energy and through their accelerometer sensors. For this, we use our current Android-based prototype of Mobile Hub and show how local data stream processing on the smartphone is used to inform whether Mobile Objects (the Sensor Tags) are in co-movement or not. This information is then sent by the smartphone over its mobile Internet connectivity to a server, executing upon our mobile middleware. A video with a preliminary version of our demo can be seen at: http://youtu.be/toEy5su_t4o.

II. A Middleware for IoMT

IoT is evolving towards a heterogenous network that combines IP-based connectivity and several short-range, last100meter wireless technologies (e.g., Bluetooth LowEnergy, NFC, ZigBee, ANT+) , the latter supported by very diverse, usualy resource-poor - and perhaps mobile - peripheral devices, from now on called smart Mobile Objects. In order to support communication in such a dynamic and heterogeneous network, we have extended our mobile cloud middleware, Scalable Data Distribution Layer (SDDL), with the generic component named Mobile Hub (M-Hub) [3]. This component runs on end-user mobile devices and opportunistically discovers and connects to nearby Mobile Objects (M-OBJs) with embedded sensors and/or actuators. The M-Hub is also capable of some processing of its own sensor and the M-OBJs’ sensor data. We further assume that the M-Hub has some means of obtaining its current geographic location.

The Scalable Data Distribution Layer (SDDL) [1], [2] is a mobile-cloud communication middleware that connects stationary DDS nodes in a cluster (or cloud) to mobile nodes that have Internet connectivity. All mobile nodes, including M-Hubs, connect to a SDDL Gateway using a reliable UDP protocol, which supports reliable message delivery in spite of intermittent wireless connectivity and handover between Gateways.

For its role as Internet-connector and local processing agent for Mobile Objects, the M-Hub requires reasonable processing power, at least one GB of memory, and network interfaces both for the Internet (e.g. 3G/4G and WiFi), and for some short-range, low-power WPAN communication technology, all of which are usually present in latest models of smart phones and tablets. Our current prototype is implemented as a set of concurrent services for Android (KitKat), and so far, supports Classic Bluetooth and Bluetooth LE as WPAN technologies. For data processing we have implemented an optional M-Hub service, the Mobile Event Processing Agent (M-EPA), a full-fledged Complex Event Processing (CEP) Esper engine2 that executes on the Android platform. With this, it is possible to program event rules and patterns using Esper Processing Language (EPL) on the smartphone - to be applied on the streams of data received from the M-OBJs. In particular, for monitoring co-movement the M-Hub continuously collects BLE-signal strength values (in dB) and 3-axis accelerometer data samples from all M-OBJs within its range, analyzes the

stream of data in time windows, and eventually generates a complex event that states that a set of M-OBJs either started co-movement or ended co-movement, and finally sends a corresponding messages to the cloud through its connection with a SDDL Gateway. The main benefits of using Esper/EPL for this data processing in the M-Hub are EPL’s rich expressiveness (with nested logic, causal, temporal and external operators) and the possibility of achieving a clear separation between the data stream analysis and sensor data acquisition and propagation issues.

Figure 1. M-Hub’s main components while it interacts with two M-OBJs

Figure 1 shows the M-Hub architecture with its Android services, and details about the interactions between them. The S2PA Service executes general tasks such as periodic wireless scans, M-OBJ registry, data transcoding, etc, for one or more WPAN technologies (currently, only Classic BT and BLE). As soon as a new M-OBJ is detected, it establishes connection, queries its sensor/actuator capabilities (in BLE) and then starts receiving sensor data updates, which are then handed over to the M-EPA. The M-EPA checks the sensor data streams received from S2PA and local sources (in this case, only the M-OBJ’s accelerometer and signal strength values) for specific patterns, according to the Complex Event Processing EPL rules set-up in M-EPA. The complex events generated by M-EPA are then handed over to the Connection Service, which attaches the current geographic position obtained from the Location Service, and sends it to the cloud via the Internet/ SDDL connection. Finally, the Energy Manager sets the periodicity/duration of WPAN scanning, location updates and Internet transmission) of the above services, in accordance with the current battery level, and if the M-Hub is plugged or not to a power source.

III. The Demo and Experimental Results

The main goal of the demo is to show a system that can detect the co-movement without any user intervention. To

2EsperTech - http://esper.codehaus.org/
accomplish it, we decided to use information such as the signal strength of the short-range technology (in our case, Bluetooth Low Energy) and accelerometer data. Since the accelerometers provides 3-axis output data, the magnitude of the accelerometer vector on the M-Obj can be calculated and compared using the equation 1.

\[ A_{mag} = \sqrt{(A_x)^2 + (A_y)^2 + (A_z)^2} \]  

(1)

Although the accelerometer data is continuously changing and is slightly different for each M-OBJ, there is also a regular, repeated pattern in similar situations with several M-OBJs, which led us to think that it would be unlikely that two M-OBJs would experience the same (or similar) accelerations unintentionally, unless they are in co-movement. A similar assumption can be made for the signal strength, since it would be very unlikely that two M-OBJs have similar signal strengths for a long period of time unless they keep an approximate same distance to each other. The previous assumptions gave us the idea of analyze the accelerometer and signal strength data of the M-OBJs on a window of time to get their average values, in order to compare them. But since the idea is to identify the co-movement in almost real-time, we chose a small window of 5 seconds. Finally, the average values obtained are compared with each other, and if they are close enough within a user defined threshold, then this means that the M-OBJs are in co-movement.

The demonstration is divided on three steps. First, we start the M-Hub service by pressing the play button located on the top-right corner of the Android application (shown on Figure 2). After that, the M-Hub will get connected with all the nearby M-OBJs and start receiving the sensor updates (e.g. Accelerometer and Signal data). Second, the server has to be running to receive the messages from the M-Hub and show them on a GoogleMap panel. Finally, we apply different movements to one or several M-OBJs in order to change their accelerometer values, so that the M-EPA service can detect when the M-OBJs are in co-movement and send the event message to the server.

A. Experiments & Conclusions

Two experiments are discussed to show the accuracy of the system. The threshold used for both experiments was 22.5 (dBm) for the signal strength, since it has a very high variation, and we only need to know if the M-OBJs are “approximately close enough”. Several tests were realized previous to the experiments in order to find a good threshold for the difference of accelerometer magnitudes, and finally, \( \Delta A_{mag} = 0.105 \) delivered the best results. On the server, we determined that the M-OBJs left the co-movement state, if an event of co-movement between two M-OBJs was not followed by another event within 11 seconds. The experiments were repeated five times.

In the first experiment a same, synchronized movement was applied to several M-OBJs (placed on a tray), and we counted the number of generated events indicating that the M-OBJs left the state of co-movement. The results showed in the table below, give the percentage of detections of non-co-movement state, when in fact the devices were in co-movement. We obtained some encouraging results, showing an 89% of success rate. For the second experiment, a different movement was applied to one of the M-OBJs, so we counted the number of times that a co-movement state was (falsely) detected. In this case, we got a success rate of 87%, with just a 12.334% of false positives.

<table>
<thead>
<tr>
<th></th>
<th>Wrong Detections (%)</th>
<th>False Positives (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>10.668</td>
<td>12.334</td>
</tr>
<tr>
<td>St.Dev.</td>
<td>3.24902</td>
<td>5.34857</td>
</tr>
</tbody>
</table>

We have presented a system to detect co-movement using the accelerometer data and the signal of a short-range wireless technology. The good results encourage us to test the M-Hub/-EPA service and its co-movement detection in real-world situations and environments, such as within a car, in a bus, etc. In the future, we shall also experiment with different definitions of co-movement (based on other data, like scalar speed) and define corresponding EPL rules to be executed on the M-Hub/-EPA.

REFERENCES