Scalable Semantic Aware Context Storage

Mário Antunes[1], Diogo Gomes[1], Rui L. Aguiar[1]

When we think about the Internet we mostly consider servers, laptops, routers and fixed broadband that have penetrated almost every household. But the fact is that the Internet is diversifying as we speak. IoT has made it possible for everyday devices to acquire and store contextual data, in order to use it at a later stage. This allows the devices to share data with one another, and even services on the Internet in order to cooperate and accomplish a given objective. A cornerstone to this connectivity landscape is machine-to-machine (M2M). M2M generally refers to information and communication technologies able to measure, deliver, digest and react upon information autonomously, i.e. with none or minimal human interaction.

Context-awareness is an intrinsic property of IoT and M2M scenarios. Context- aware communications and computing has played a critical role in understanding sensor data. In Iot/M2M scenarios, an entity's context can be used to provide added value: improve efficiency, optimize resources and detect anomalies. The following examples illustrate the importance of context information in M2M scenarios. Fusing data from several sensors makes it possible to predict a driver's ideal parking spot. Projects such as Pothole Patrol and Nericell use vehicular accelerations to monitor road conditions and detect potholes. TIME (Transport Information Monitoring Environment) project combines data from mobile and fixed sensors in order to evaluate road congestion in real time.

However, recent projects follow a vertical approach, devices/manufacturers cannot share context information because each one uses a different structure, leading to information silos. This has hindered interoperability and the realisation of even more powerful IoT and M2M scenarios.

Common definitions of context information are so broad that any data related to an entity can be considered context information. These definitions also do not provide any insight about the structure of context information. Currently there is no uniform way to share/understand vast amounts of IoT/M2M data. Usually, each context-aware platform defines its own context representation based on the platform requirements. This breaks compatibility between platforms and limits the quantity of context information that can be used in M2M applications, impairing future developments.

It is unlikely that in the future a context representation standard will be widely adopted. First, the diversity of context representations, each one of them was designed for a specific usage and/or data types. Second, a widely adopted context representation does not completely solve the issue of knowledge extraction. Due to the vast amount of data it is extremely difficult to define a priori all the relations between information sources, patterns, and even possible optimizations.

In order to deal with these issues, we de ned the basic context storage requirements, analysed the impact of context organization models and proposed a new context organization model for generic IoT/M2M applications. The simplest way to model context information is through a 1-dimension model (see Figure 1). Each piece of data is characterized by a single key, stored and indexed individually.

The 1-dimension model leads to poor performance and scalability. We devised an d-dimensional model (see Figure 2) to overcome the limitations of the former. Instead of storing documents independently, they are organized by device id. The platform stores all the documents, but only needs to index the sources. The remaining d – 1 dimensions are used to filter data from a specific source. Our organization model was evaluated with a stress simulation based on three M2M projects, outperforming the competition.

1 — Department of Electronics, Telecommunications and Informatics & IT, University of Aveiro

FIGURE 1
Representation of a 1-dimension field

FIGURE 2
Representation of a d-dimension model, with 2 dimensions. The first and second dimensions are device id and time respectively.