



M2M: machine to machine. Right? Partially

Since the emergence of the M2M concept, society has gradually been getting the impression there is some kind of new world, a world of machines that is separate from the world of people. It is as if we had been invaded by an army of millions of alien machines that live their lives in isolation, awaiting the moment to launch the final assault and replace us.

The figures are dizzying. There are already many more machines than people, and the proportion will continue to shoot up at the same time as they all connect to each other to form an unimaginable super-organism.

This conception of both worlds, that of people and that of machines, almost antagonical, is profoundly mistaken.

The key word to understand the relationship between people and machines is "purpose". Each machine has a purpose at the service of people and each increase in the number of devices and the quality of the communication between them also has a purpose: to improve that service.

Service is expressed in many ways, sometimes allowing the effortless achievement of tasks that would be impossible for a human alone and sometimes improving the efficiency of business processes (doing more with less). Thus, "efficiency" is another key word in the world of M2M.

If there were no need to worry about the time and the amount of resources needed to achieve a particular purpose, M2M would not exist. Why set up two machines to communicate with each other if we can put a couple of buttons on each one and go from one to the other pressing the buttons and getting them to do what they have to do at the right time and in the right order? Because doing it that way would be slow and expensive (people are very expensive, that's why devices usually live alone).

We have a pile of machines that achieve a purpose when they carry out their actions in an orderly manner and we need them to work efficiently in terms of time and cost. At this point, M2M is born for, in fact, we could change its name and call it E&C, efficiency and control.

Efficiency to achieve purposes using few resources and control to know that everything is going well and to implement corrective actions if not.

What is a machine?

If we want to understand what M2M means we have to understand what the emms of M2M mean.

Do we know what machines are?

We may think so but in fact we don't, and the reason is because there is a blurry frontier on which machines and the people interacting with them merge together.

No-one would doubt that a toaster or a coffee maker is a machine, but what about an iPad or a laptop? Are they machines? Many would say they are not, partly because they are intimately linked to people (people use them). It is as if a machine has to have gears and an engine to be acknowledged as such.

I wouldn't like to get tangled up in definitions; in fact, if you are curious enough to look the word up in a standard dictionary, you can confirm the wide variety of objects included with the concept of "machine". In the final analysis, it doesn't really matter what a machine is, what matters is that there is a common understanding of what it is in the context in which it is mentioned. What is a machine in the context of M2M?

An M2M machine is an object with a purpose and is capable of communicating with other objects to receive or transmit data (by the way, an order is a special type of data that triggers an action in the object receiving it, but it is still data; everything is data).

To make it even easier, let me tell you a joke: an M2M machine does something because something else tells it to, and at the end of all this chain of somethings there is a person (who is just another kind of something ...).

The shape of M2M

If I had wanted to appear more elegant than I am, I should have said "The topology of M2M" but, once more, I stress that we must not get lost in words; rather, we must concentrate on the concepts and their meaning.

In an M2M network, machines communicate with each other, but ... are there a lot of machines or only a few? Are they near or far? Do they talk all the time or just say "hi" now and again?

The answer to all these questions is a tremendous constraint on the technical solutions implemented in the world of M2M. There is no universal answer; each answer depends on the question.

The most common question or, to put it another way, the most common use case to be resolved is that of a large number of isolated machines distributed over a very extensive geographical area and needing to communicate with a central machine (the server) to send data or receive orders. This concept can be applied to electricity meters, vehicles, coffee makers or anything else.

The typical "shape" of M2M is therefore that of a star: a centre surrounded by thousands or millions of machines (that can also talk to each other, although this latter case is less common).

Does the shape change? Yes, in two different ways, one because machines appear and disappear and the other because machines can move. An important issue determining in part the dynamic nature of an M2M system is whether or not the machines are associated directly to people; in most cases, they aren't. Unlike what happens with a cell phone, they are independent of their users, and even of

the place where they are located. For example, a coffee maker may not be connected to the communications infrastructure of the building it is in, because the machine belongs to the company that is using it and the building to another different company.

This characteristic of M2M services, marked by a strong spatial dispersion, because they are so massive, by the need to communicate independently and by their mobility, means that most of the M2M communications solutions being rolled out around the world are based on the use of mobile data networks, which have the great logistic advantage that machines already incorporate their own factory-set communications system.

With all this, the typical "shape" of M2M would be that of a star with a very powerful centre, and with a huge, variable number of points that may or may not be moving.

The layers of an M2M service

Machines have a purpose and to achieve that purpose they need to communicate with the servers controlling them (which are also machines).

A Point of Sale (PoS) terminal has to communicate with the bank's server to be able to accept a card payment. An electricity meter has to communicate with the server of the utility company to report on the consumption for the bill to be calculated. A car has to communicate with the car manufacturer's server to report on the status of the vehicle or a breakdown.

In all these processes for communication between machines, there are three distinct layers of service, each of which revolves around a different concept:

1. **Communications** layer: in view of the preponderance of mobile networks in the realm of M2M, this layer, which is in charge of providing communication between machines, revolves around a mobile system, although other technologies can also be used.
2. **Machine** layer (devices): machines are physical hardware. They are the stars of M2M because they carry out the actions needed to achieve the purpose: collect information, process it, transmit it or implement control actions.
3. **Application** data layer: this contains the data exchanged between machines. The purpose of this flow is different in each specific case (data from sensors, the car engine, a credit card, an irrigation connection, etc).

The set of elements making up the three layers of M2M services needs to be managed, i.e. the operations needed to make everything work correctly have to be carried out along with the actions for which they have been designed.

As a result, the essential concept in M2M services is: control. Control can be translated as the management of the service provided by the machines, and exercising this control requires knowledge of the status of the M2M system as a whole and on all three layers: communications, machines and applications (data).

Each of the service layers has to operate correctly. If any of them fails, it may not be possible to provide the service, so control over each one is essential. All of them must be visible and all controllable.

Black Box, Transparent Box

M2M services provide transparency and control.

Let's imagine for a moment that someone comes knocking at our door and puts us in charge of a service requiring huge numbers of machines spread all around the world. The service can be anything you like, but whatever it is, it will have three layers: communications, machines and applications (data).

How do we solve the problems of transparency and control on each layer?

Controlling the communications. Good bye, black box

The service is a black box when we know nothing about what goes on inside it and, therefore, whether or not it is operating correctly. Things go into a black box and we hope the outcomes we want will come out. If something goes wrong, we may or may not know about it, and we may find out about it sooner or later depending on how serious the problem is and how evident its impacts are on the expected output.

This is a very dangerous situation, because if a problem arises in any business-critical process, the time we take to resolve it may cause considerable losses. To avoid this, first of all, we have to be sure that the communications are working well between the machines, that they are sending the information at the right time, that they have not been hacked, and that we can control the costs associated with those communications.

All this control doesn't have to do with the machine itself, nor with the application running on it, but with the simple ability to communicate: if there is no communication, the process fails even if the machine is working.

This problem can be solved by using an M2M platform that makes the communication status of all the machines visible, with all their connectivity logs, the expense incurred and even their location. This will not tell us if the service is working correctly or not (only in part), but we will know that it may be working, because the machines are able to communicate.

With this information in our hands, we will have turned the black box into one with some parts that are transparent and others that are not.

Controlling the machines. Learning to take care of the box

Once the communications are up and running, the data can flow between the machines and the servers, and the system can fulfil its function.

We have uncovered part of the box by making some things transparent and we can see that there are IP addresses that we can reach, telephone numbers we can text messages to ... IP addresses? Interesting, but what machine is at that IP address? A car? A train? Or the door to a garage? Is it old or new? What model? What operating system is it using?

Answering questions like these about the machine is necessary to manage its life cycle, to know when it has to be updated, or even when something has broken down and it needs to be replaced, entirely or in part.

A machine's life cycle starts when it is manufactured and ends when it is decommissioned, and there may be hardware or software changes made to it, repairs or updates. By managing machines' life cycles, understanding how healthy they are, and doing so remotely as far as possible, is a requirement for an M2M system (efficiency again).

This part of the system is covered by two concepts: I+M and DMM (Installation & Maintenance and Device Management and Monitoring).

An M2M system has to be able to handle both mobile lines (communications) and machines and to make both of these visible, as well as the way they are inter-related.

Once we have reached this point, more than half the box has become transparent, and we even have a few buttons available to act on it remotely, so we are moderately happy and can sleep better.

We know that our machines are visible and communicating, and we know what they are; if they are blue or yellow, large or small, old or new.

We are closer to the transparent box, we only have one thing left: is our blue box at IP address 1.3.4.5 with version 2.0 of the operating system doing what it's supposed to? Oops ... it looks like we don't know.

Controlling the information. Transparent box: tell me what you know.

Machines have a purpose, and to achieve that purpose, they generally collect information from their surroundings and send it to their servers. This information is extremely diverse: electricity consumption, the temperature in a room, the number of doughnuts left in a vending machine, etc.

Machines have applications, the programmes that collect, store and transmit these data. Some data are tremendously specialized and cannot be processed with a general routine (for example, credit card transactions through a PoS Terminal) but in many other cases the data are collected and used in a very similar way, regardless of their exact nature.

There are two ways of approaching the design of solutions for the data layer of M2M services: build a specific service for that use case or take advantage of a horizontal M2M platform that solves the common issue.

This common problem is the ability to interact with machines to send or receive data, store the information they provide in a highly flexible database and enable mechanisms for external consumers (people at the end of the day) can access that information, discover any changes in it and use the data for whatever their purpose is. We normally call this component by a very clear name, Data Collection and Analysis (DCA).

Of course, DCA on its own does not constitute the entire service. It is only a part. It always needs to be customized for each use case involved. This specialization is based on two aspects: the creation of connectors speaking the right language for each machine we want to include and the creation of an application to take in the data stored and do something specific with them (reports, alerts, billing or anything else).

Ideally, all the information on all three layers of the service (connectivity, devices and application

data) have to be shown in a joint, co-ordinated way, but filtered into different views, associated with the types of user accessing the information and their individual needs.

Now we have our transparent box and we can finally sleep easy despite being in charge of a million machines. It is not that they are never going to fail, of course they will, but when that happens, we will know about it and we will be able to put it right. Uncertainty is the real killer.