CHAPTER

Real-World Design Constraints

9

CHAPTER OUTLINE

9.1 Introduction

This chapter outlines the technical design constraints to illustrate the questions that need to be taken into account when developing and implementing M2M and IoT solutions in the real world. This provides some background and thoughts for the use cases outlined in Part III of this book.

9.2 Technical design constraints – hardware is popular again

The IoT will see additional circuitry built into a number of existing products and machines – from washing machines to meters. Giving these things an identity, and the ability to represent themselves online and communicate

From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence.
DOI: http://dx.doi.org/10.1016/B978-0-12-407684-6.00009-7

with applications and other things, represents a significant, widely recognized opportunity.

For manufacturers of products that typically contain electronic components, this process will be relatively straightforward. Selection of appropriate communications technologies that can be integrated with legacy designs (e.g. motherboards) will be relatively painless. The operational environments and the criticality of the information transmitted to and from these products, however, will present some unconventional challenges and design considerations. These are discussed later in the context of new and potential applications.

The IoT will, on the other hand, allow for the development of novel applications in all imaginable scenarios. Emerging applications of M2M and wireless sensor and actuator networks have seen deployment of sensing capabilities in the wild that allow stakeholders to optimize their businesses, glean new insight into relevant physical and environmental processes, and understand and control situations that would have previously been inaccessible.

The technical design of any M2M or IoT solution requires a fundamental understanding of the specificity of the intended application and business proposition, in addition to heterogeneity of existing solutions. Developing an end-to-end instance of an M2M or IoT solution requires the careful selection, and in most cases, development of a number of complementary technologies. This can be both a difficult conceptual problem and integration challenge, and requires the involvement of the key stakeholder(s) on a number of conceptual and technological levels. Typically, it can be considered to be a combinatorial optimization problem — where the optimal solution is the one that satisfies all functional and non-functional requirements, whilst simultaneously delivering a satisfactory cost-benefit ratio. This is particularly relevant for organizations wishing to compete with existing offerings, or for start-up ventures in novel application areas. Typically, capital costs in terms of "commissioning" and operational costs in "maintenance" must be considered. These may be balanced by resultant optimizations.

Typical M2M or IoT applications conform to the general functional architecture presented in Chapters 6, 7, and 8. Assuming that the system designer has selected the appropriate communications technologies to bridge the device and application domains (likely standard Internet Protocol (IP)-based methods as described in Chapter 5), he or she must consider the application at several levels: the device (or M2M Area Network; i.e. hardware), representation (i.e. data and visualization thereof), and interaction (i.e. local or remote control).

9.2.1 Devices and networks

Introduced in Chapter 5, devices that form networks in the M2M Area Network domain must be selected, or designed, with certain functionality in mind. At a minimum, they must have an energy source (e.g. batteries, increasingly EH), computational capability (e.g. an MCU), appropriate communications interface (e.g. a Radio Frequency Integrated Circuit (RFIC) and front end RF circuitry), memory (program and data), and sensing (and/or actuation) capability.

These must be integrated in such a way that the functional requirements of the desired application can be satisfied, in addition to a number of non-functional requirements that will exist in all cases.

9.2.1.1 Functional requirements

Specific sensing and actuating capabilities are basic functional requirements. In every case — with the exception of devices that might be deployed as a routing device in the case of range issues between sensing and/or actuating devices — the device must be capable of sensing or perceiving something interesting from the environment. This is the basis of the application. Sensors, broadly speaking, are difficult to categorize effectively. Selecting a sensor that is capable of detecting a particular phenomenon of interest is essential. The sensor may directly measure the phenomenon of interest (e.g. temperature), or may be used to derive data or information about the phenomenon of interest, based on additional knowledge (e.g. a level of comfort). Sensors may sense a phenomenon that is local (i.e. a meter detecting total electricity consumption of a space) or distributed (e.g. the weather).

In many cases, sensing may be prohibitively expensive or unjustifiable at scale, and thus motivates the derivation of models that can reason over the sensor readings that are available. Air and water quality monitoring systems are typical of this type of problem.

Given a particular phenomenon of interest, there are often numerous sensors capable of detecting the same phenomenon (e.g. types of temperature sensors), but have widely varying characteristics. These characteristics relate to the accuracy of the sensor, its susceptibility to changing environmental conditions, its power requirements, its signal conditioning requirements, and so forth.

In some cases, for example, a complementary (e.g. temperature) sensor is required in addition to the primary sensor such that variations in readings of the primary sensor that are caused by variation in temperature can be understood in context.

Sensing principle and data requirements are also of essence when considering the real-world application. Consider a continuously sampling sensor, such as an accelerometer, versus a displacement transducer. Displacement can be sampled intermittently, whereas if an accelerometer is duty-cycled, it is likely that data points of interest (i.e. real-world events) may be missed. Furthermore, the data requirements of the stakeholder must be taken into account. If all data points are required to be transmitted (which is the case in many scenarios, irrespective of the ability to reason locally within an M2M Area Network or WSN), this implies higher network throughput, data loss, energy use, etc. These requirements tend to change on a case-by-case basis.

9.2.1.2 Sensing and communications field

The sensing field is of importance when considering both the phenomenon to be sensed (i.e. Is it local or distributed?) and the distance between sensing points. The physical environment has an implication on the communications technologies selected and the reliability of the system in operation thereafter. Devices must be placed in close enough proximity to communicate. Where the distance is too great, routing devices may be necessary. Devices may become intermittently disconnected due to the time varying, stochastic nature of the wireless medium. Certain environments may be fundamentally more suited to wireless propagation than others. For example, studies have shown that tunnels are excellent environments for wireless propagation, whereas, where RF shielding can occur (e.g. in a heavy construction environment), communication range of devices can be significantly reduced.

9.2.1.3 Programming and embedded intelligence

Devices in the IoT are fundamentally heterogeneous. There are, and will continue to be, various computational architectures, including MCUs (8-, 16-, 32-bit, ARM, 8051, RISC, Intel, etc.), signal conditioning (e.g. ADC), and memory (ROM, (S/F/D)RAM, etc.), in addition to communications media, peripheral components (sensors, actuators, buttons, screens, LEDs), etc. In some applications, where it would previously have been typical to have homogeneous devices, a variety of sensors and actuators can actually exist, working collaboratively, but constituting a heterogeneous network in reality.

In every case, an application programmer must consider the hardware selected or designed, and its capabilities. Typically, applications may be thought of cyclically and logically. Application-level logic dictates the sampling rate of the sensor, the local processing performed on sensor readings, the transmission schedule (or reporting rate), and the management of the communications protocol stack, among other things. Careful implementation of the (embedded) software is required to ensure that the device operates as desired. This continues to be non-trivial and highly specialized. For heterogeneous devices, the embedded software will vary by device.

The ability to reconfigure and reprogram devices is still an unresolved issue for the research community in sensor networks, M2M, and the IoT. It relates both to the physical composition of devices, logical construction of the embedded software, and addressability of individual devices and security, to name a few. Operating systems are typically used to make programming simpler and modular for embedded systems designers, but each comes with conceptual and implementation differences that impact the ability to handle certain desirable features.

9.2.1.4 Power

Power is essential for any embedded or IoT device. Depending on the application, power may be provided by the mains, batteries, or conversion from energy scavengers (often implemented as hybrid power sources). The power source has a significant implication on the design of the entire system. If a finite power supply is used, such as a battery, then the hardware selected, in addition to the application level logic and communications technology, collectively have a major impact on the longevity of the application. This results in short-lived applications or increased maintenance costs. In most cases, it should be possible to analytically model the power requirements of the application prior to deployment. This allows the designer to estimate the cost of maintenance over time.

9.2.1.5 Gateway

The Gateway, described in Chapter 5, is typically more straightforward to design if it usually acts as a proxy; however, there are very few effective M2M or IoT Gateway devices available on the market today. Depending on the application, power considerations must be taken into account. It is also thought that the Gateway device can be exploited for performing some level of analytics on data transitioning to and from capillary networks.

9.2.1.6 Nonfunctional requirements

There are a number of nonfunctional requirements that need to be satisfied for every application. These are technical and non-technical:

- Regulations
  - For applications that require placing nodes in public places, planning permission often becomes an issue.
  - Radio Frequency (RF) regulations limit the power with which transmitters can broadcast. This varies by region and frequency band.

- Ease of use, installation, maintenance, accessibility
  - Simplification of installation and configuration of IoT applications is as yet unresolved beyond well-known, off-the-shelf systems. It is difficult to conceive a general solution to this problem. This relates to positioning, placement, site surveying, programming, and physical accessibility of devices for maintenance purposes.
- Physical constraints (from several perspectives)
  - Can the additional electronics be easily integrated into the existing system?
  - Are there physical size limitations on the device as a result of the deployment scenario?
  - What kind of packaging is most suitable (e.g. IP-rated enclosures for outdoor deployment)?
  - What kind and size of antenna can I use?
  - What kind of power supply can I use given size restrictions (relates to harvesting, batteries, and alternative storage, e.g. supercapacitors)?

9.2.1.7 Financial cost

Financial cost considerations are as follows:

- **Component Selection:** Typically, the use of these devices in the M2M Area Network domain is seen to reduce the overall cost burden by using non-leased communications infrastructure. However, there are research and development costs likely to be incurred for each individual application in the IoT that requires device development or integration. Developing devices in small quantities is expensive.
- **Integrated Device Design:** Once the energy, sensors, actuators, computation, memory, power, connectivity, physical, and other functional and nonfunctional requirements are considered, it is likely that an integrated device must be produced. This is essentially going to be an exercise in Printed Circuit Board (PCB) design, but will in many cases require some consideration to be paid to the RF front-end design. This means that the PCB design will require specific attention to be paid to the reference designs of the RFIC manufacturer during development, or potentially the integration of an additional Integrated Circuit (IC) that deals with the balun and matching network required.

9.3 Data representation and visualization

Each IoT application has an optimal visual representation of the data and the system. Data that is generated from heterogeneous systems has heterogeneous visualization requirements. There are currently no satisfactory standard data representation and storage methods that satisfy all of the potential IoT applications.

Data-derivative products will have further *ad hoc* visualization requirements. A derivative in these terms exists once a function has been performed on an initial data set – which may or may not be raw data. These can be further integrated at various levels of abstraction, depending on the logic of the integrator. New information sources, such as those derived from integrated data streams from various logically correlated IoT applications, will present interesting representation and visualization challenges.

9.4 Interaction and remote control

To exploit remote interaction and control over IoT applications, connectivity that spans the traditional Internet (i.e. from anywhere) on the side of the application manager, or other authorized entity, to the end-point (i.e. an embedded device), continues to be a challenging problem. Aside from authentication and availability challenges, for most constrained devices, heterogeneous software architectures, such as event-based operating systems running on devices with significantly varying concurrency models, continue to pose significant challenges from a remote management perspective.

Elements of Device Management, specifically reprogramming and reconfiguration of deeply embedded devices, will be required, particularly for devices deployed in inaccessible locations. This requires, among others, reliability, availability, security, energy efficiency, and latency performance, to be satisfactory whilst communicating across complex distributed systems.

Another significantly under-researched topic is the definition and delivery of end-to-end quality of service (QoS) metrics and mechanisms in IoT-type applications. These will be necessary if Service Agreements (SA) or Service Level Agreements (SLA) are to be defined in the case of service provisions for IoT applications – which may or may not be desirable to the application owner. This will be situation-specific. End-to-end latency, security, reliability, availability, times between failure and repair, responsibility, etc., are all likely to feature in such agreements.