Digital twins for Industrial IoT with Altior

A technical white paper

Part 1: digital twin design

InkwellData
Connected Innovation

Executive summary

Inkwell Data Limited is an Industrial "Internet of Things" (IoT) connectivity provider, focused on delivering easy-to-use, secure, robust, scalable and cost competitive solutions.

We believe that the technological implementation of "digital twins" will become instrumental in unlocking the value in Operation that the Internet of Things can bring.
Digital twins have been around for more than a decade, but are typically highly tailored and complicated, requiring extensive software development time and skills.

Connecting industrial objects at an enterprise scale requires a simpler, more standardised approach, without losing any of its efficacy.

Through its novel middleware IoT platform, Altior, Inkwell Data Ltd is delivering such a new approach.

This paper is the start of a series that will explain how the author, Massimo Cesaro, co-founder and CTO of Inkwell Data Ltd, has created such a platform and ecosystem.

Digital twin for Industrial IoT with Altior

Introduction

The purpose of this series to introduce the reader to the Altior architecture and how it can be applied to a wide range of different IoT use cases, without requiring a specific knowledge of software development for telecommunication engineering disciplines.

About the digital twin

The concept of digital twins was originally envisioned in 1991 [1], although it took ten years for the concept (under a different name) to be implemented for product lifecycle management ("PLM") purposes [2]. The "digital twins" moniker was officially used in 2010 by the NASA [3] and has since become popular outside the PLM field, adapted to many other industries. Although many commentators tend to overload the definition of digital twins, the concept is actually intuitive.

For the purpose of this white paper, we will adopt the definition of a digital twin as "a digital model of a physical object and its associated processes and systems". A digital twin is therefore a digital replica template of both a real device (such as a sensor or a meter), as well as all the processes/interactions of the device with the real world.

About the IoT and Industrial IoT

Just like the digital twin definition, the definition of the "Internet of Things" varies widely.

The first reference to the Internet of Things is commonly dated to 1999, with an MIT presentation about RFID [4], although the concept of interconnected objects and devices actually goes back by at least 30 years.

A generally agreed connotation of IoT is a group of "sensors and actuators embedded in physical objects which are linked through wired and/or wireless networks, typically using the same Internet Protocol (IP) to connect to the Internet".

Over the years, many other names have been applied to the same concept, such as "M2M" (machine to machine) communication or the German popular term, "Industry 4.0". Still, it boils down to having physical devices communicating with applications via some form of connectivity. For the purpose of this paper we will focus on a specific area, the Industrial Internet of Things ("IIoT") [5], which addresses industrial sensors and applications (B2B), rather than consumers applications (B2C).

Joining forces

For the rest of this document, we will refer to a digital twin as a "digital dynamic representation of a physical system, which is connected to it over the entire lifecycle for bidirectional data exchanges".

Creating digital twins for IoT applications usually means defining a digital model for one or more specific sensor, set up a communication network to receive and send data to and from the sensors, and develop a software application that handles the actual data processing[1].

The success of a digital twin implementation therefore depends on multiple variables, including the actual size and complexity of the physical and logical entity to be replicated, as well as the level of accuracy required by the user.

For non-trivial systems, developing a digital twin can be a time-consuming task, requiring significant planning as well as a stable baseline; changes to the physical infrastructure can adversely impact the efficacy of the digital twin.

Creating digital twins on Altior

Altior by Inkwell Data is a one-stop-shop digital twin development technology that provides all that is required to integrate the physical asset with its digital twin, to enable real-time flow of data from the IIoT device, as well as integration of this data with operational transactional information from other enterprise systems.

Altior enables the rapid design of a digital twin, starting with the "mapping" of the features of the physical object to their equivalent digital version, called digital twin "properties". Properties are identified by name, a data type and the kind of interactions they support (read/write or get/set).

[1] For the purpose of this document we differentiate the digital twin from a digital simulation, as normally the scope of a simulation differs from that of an IIoT digital twin.

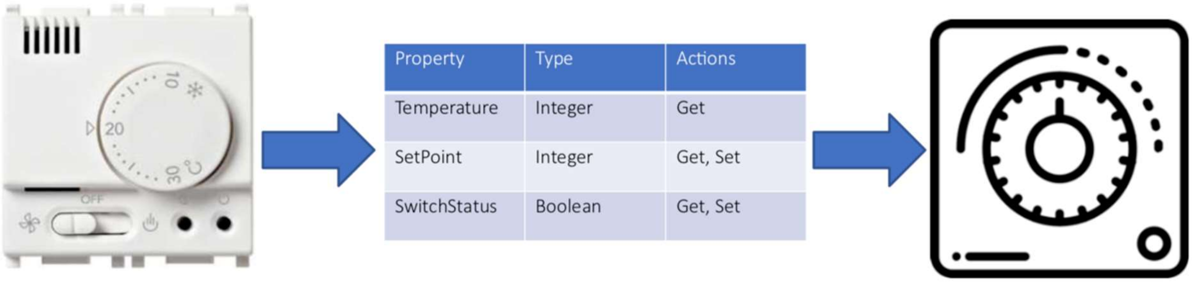

Figure 1 – Mapping a physical object to its digital twin

Using the example of the thermostat, in figure 1, defining its digital twin to map physical features to their virtual counterpart on Altior is a straightforward process:

Physical Object	**Digital Twin Property**		
Feature	*Name*	*Data type*	*Interaction type*
Temperature sensor	Temperature	Numeric	Read
Temperature setting	Setpoint	Numeric	Read, Write
Switch	Switch status	Boolean (on, off)	Read
Battery	Battery status	Numeric	Read
Serial Number	Serial number	Alphanumeric string	Read

Table 1 – A simple thermostat device twin definition

Extending the digital twins

By using the simple data structure definition above, Altior enables the creation of a device twin for collecting data and operating a real thermostat, over any supported network technology.
Reading the digital twin “Temperature” property from the device twin means getting the current value of the temperature measured by the temperature sensor on the physical object. Writing on the “Setpoint” property in the device twin allows to set the switching temperature on the physical object.

In both cases (reading and writing properties), the communication process to and from the physical object is transparent and handled behind the scene by Altior (see part 2).
To the physical object basic attributes defined manually by the device twin designer, Altior by default adds to every digital twin a few other base properties that are useful for the object’s management, including:

Digital Twin Property	Purpose
Unique ID	Uniquely identify a digital twin instance
GPS coordinates	Current geographical position of the device twin
Security provider	Identifies the security mechanism used for the device communication
Version	The device twin version (that may vary for the same kind of object)
Manufacturer	The name of the manufacturing company of the physical device
Network adapters	A list of communication network types used for the device communication

Table 2 – A selection of Altior device twin basic device properties

The actual digital twin definition might include many other attributes, such as an image or a photo of the physical object for reference, as well as additional communication security features and one or more device manufacturer's custom attributes and properties (for example the object's firmware version). The digital twin designer can even add complex data structures such as lists and maps to the digital twin definition, as there are few, if any, limits on the data types managed by Altior.

The device digital twin design process on Altior is made simple by using an intuitive web-based user interface that guides the digital twin designer through the easy steps of the properties' definition. This will create a digital twin "template", that acts like a mold, a form, to create digital twins.

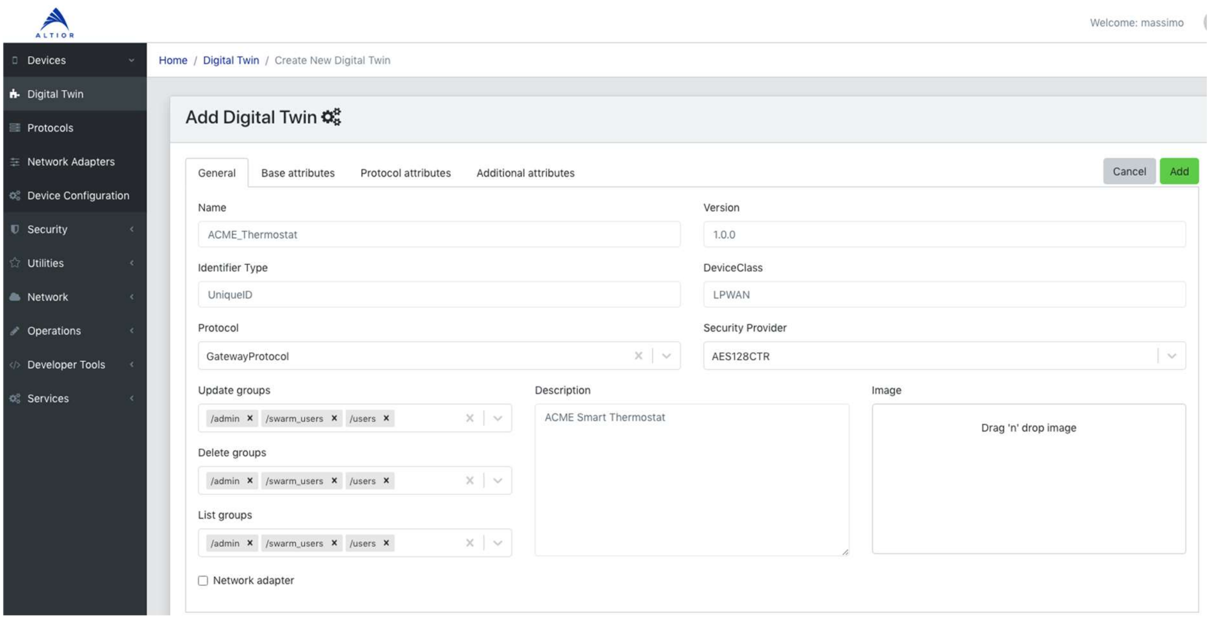

Figure 2 – Designing a device twin

Once the digital twin template definition is complete and validated, it is placed on a repository where it can be re-used for deriving other digital twin templates for similar objects.

The template is used to generate the software for the digital twin; code generation and deployment on an Altior server simply involves invoking the "builder" function from the Altior front end.

The Altior builder takes as input the digital twin definition and automatically emits as output the formally correct source code of the device twin implementation. The source code is then complied in the executable format and deployed to the Altior runtime service server for managed execution.

Adding security and communication

While most of the device twin base properties listed in table 2 are self-explanatory (such as the GPS coordinates), a couple of them deserve a closer look.

The **security provider** property is the name of the digital twin security model provided by *Aegis*, Altior's security framework; a security provider implements the security strategies for the single device twin.

An Aegis security provider is a plug-in software component that is invoked every time there is a security related operation on the device twin.

Among the features exposed by an Aegis security provider there are the encryption and decryption algorithms used to exchange raw (binary) data on unsecured networks, or the functions to access a key management server.

Aegis offers some popular security algorithms and strategies; however, the users can implement their own proprietary strategies using the Aegis API.

In Altior, the effort to ensure data security is part of the building process of the digital twin. Any device twin generated on Altior has a basic security provider that can be extended to support multiple security layers.

Altior device twins are subject to security checks for every operation they are involved in; just like Altior users have login credentials and a privilege set, the device twin instances have their own particular credentials and privileges.

Device twin actions can be singularly restricted in what they do and how they do it, and this is a way to ensure that compromising a device twin does not meant compromising the entire Altior system.

The network adapters property is a list of Altior modules used by the device twin to communicate on physical networks.

Device twins can exchange data with their physical counterparts when connected through a network infrastructure.

Depending on the nature of the infrastructure, the device twin access to the communication layer can vary.

For example, access to a device from an LPWAN network communication is usually different from accessing it from a cellular network; assigning one or more network adapters to a device twin is a way to allow communication of the virtual device to the real world in a simple and consistent manner.

Working with Altior device twins

Once the definition of the digital twin template is complete and validated, Altior is ready to deal with the physical objects that will be connected to their digital counterparts.

The first step is to create an instance of the digital twin that will be linked to the physical object it represents.

Device twin instances are created from their base template by assigning one or more property values to identify the associated physical object.

This task can be manually performed by a system administrator from the Altior web front end:

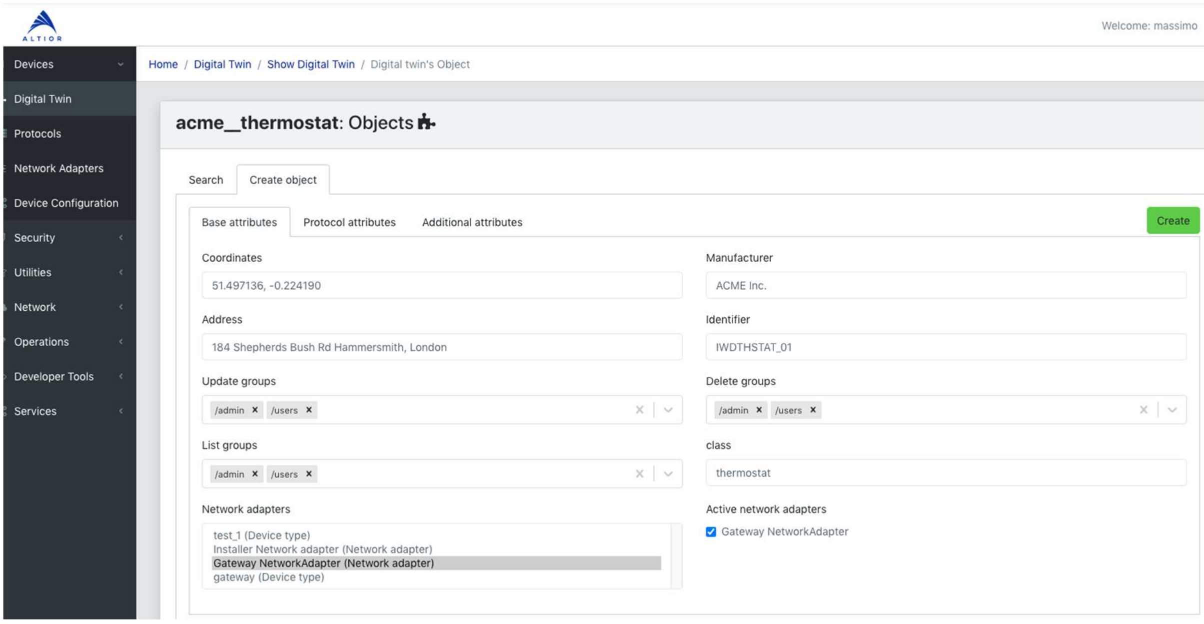

Figure 3 – Creating device twin instances

In the example above, Altior will create a device twin instance with the unique identifier IWDTHSTAT_01 to communicate with a specific physical thermostat.

With the same identifier (for example the serial number), using an LPWAN network protocol and through an IoT gateway, that can be as simple as a common Wi-Fi access point or a dedicated LPWAN gateway.

> While manually creating device twin instances is useful for prototyping IoT applications or for creating a small number of device twins, in real world applications where the number of physical objects can be large, Altior allows the batch creation of many device twins instances in a single operation, either by importing their definition from an external file or by using the Altior device twin application programming interface (API) for an external application.

The device twin instance is connected to the corresponding physical object using a network adapter that creates a bridge between the two worlds.

Once it is "created" on Altior, the device twin instance becomes a faithful copy of the physical device state. Every state modification from one side is reflected on the counterpart.

In our thermostat example, if the room temperature changes then the temperature sensor property of the device twin is updated as soon as possible[2].

Conversely, if an Altior application changes the "Setpoint" property of the device twin, the new setting is sent to the physical thermostat.

The RTE: a separate reality

The device twin instances are basically the state representation of the associated physical objects.

They are created and maintained in an Altior subsystem called "runtime environment" or RTE.

The RTE is a process container for all the device twin instances, regardless of their type: the RTE, the thermostat device twins and, for example, the smart meter device twins are kept together, although the rules to access their state information are completely different and they are connected to different networks.

In fact, every device twin instance is an isolated lightweight process independently managed by the RTE.

This means that a fault in a single device instance will not affect any other instance; moreover, if the fault is recoverable the RTE will automatically restart the failed device twin instance.

[2] Ideally the state update should happen in real time, and Altior supports this model. However, due to physical constraints like a network temporary unavailability or unexpected latency, the actual status update should be considered as "best effort".

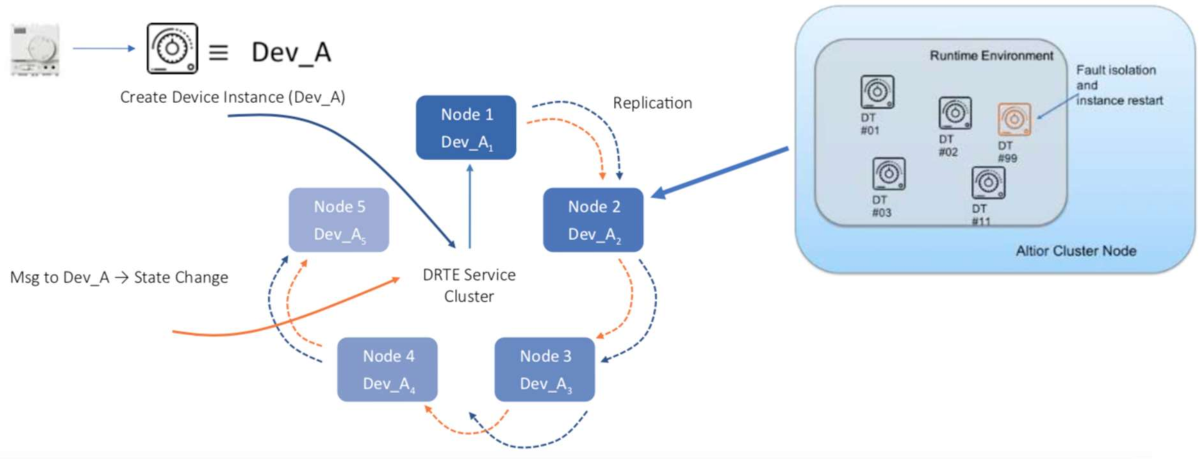

Figure 4 – The Runtime Environment RTE

As such, the RTE allows the management of millions of device twins instances, on commodity server hardware.

Altior makes good use of available computing resources, mainly the CPU power and the RAM memory and allows for a simple vertical scalability by just adding more resources to a running system.

The RTE can also be transparently replicated on one or more different servers, creating an RTE cluster and providing horizontal scalability.

Thanks to the RTE, it is possible to avoid a single point of failure in the system by replicating the data into different interconnected servers.

Altior clustering works for every Altior function since it is not limited to the RTE.

The Altior cluster setup is transparent to the user and it does not require any external support in form of a specific cloud infrastructure or services, making Altior's distribution and replication cloud-agnostic.

RTE clustering is flexible and can be also used for load balancing of the device twin status updates among different servers, or to support peak requests in consequence of an external event (such as a power supply black-out for example).

Device twin instances in the RTE are ephemeral data structures reflecting the current state of real objects, and as such they do not require to keep track of the changes over time. This makes the Altior RTE extremely responsive, even under heavy load, and for a large number of objects managed.

However, in situations where state change persistence is required, Altior offers many options to implement application-specific storage strategies using the Altior API or the Altior application deployment tools, that will be subject of Part 3 of this series.

Conclusions

The Altior platform is an enabling technology for the development of Industrial IoT applications using the digital twin concept.

The device twin abstraction available on Altior makes it possible to model not only physical devices, but communication networks and business processes.

In the next part of this series, the Altior middleware features will be introduced.

We will look into Altior's technology agnostic approach to bringing the physical world to the digital twins, and the way Altior allows building heterogeneous device twin networks by abstracting the communication interfaces.

References

[1] Gelernter, Hillel (1991). Mirror Worlds: or the Day Software Puts the Universe in a Shoebox—How It Will Happen and What It Will Mean. Oxford; New York: Oxford University Press. ISBN 978-0195079067.

[2] Grieves, Michael. (2016). Origins of the Digital Twin Concept. 10.13140/RG.2.2.26367.61609.

[3] Piascik, R., et al., Technology Area 12: Materials, Structures, Mechanical Systems, and Manufacturing Road Map. 2010, NASA Office of Chief Technologist.

[4] Ashton, K. (22 June 2009). "That 'Internet of Things' Thing, https://www.rfidjournal.com/that-internet-of-things-thing

[5] Boyes, H., et al., "The industrial internet of things (IIoT): An analysis framework", Computers in Industry, Volume 101, 2018, Pages 1-12, ISSN 0166-3615

For more information and contact requests:

Inkwell Data Ltd		Inkwell Data Italia
Iveagh Court		
Harcourt Road	184, Shepherds Bush Road	via degli Zabarella, 24
Dublin 2 - Ireland	W6 7NL London - UK	35121 Padova - Italy
+44 (0)203 951 0831	+44 (0)203 951 0831	+39 (0)49 689 8423

info@inkwelldata.com https://inkwelldata.com

Cover photo by Eric Krull on unsplash.com

