Getting Started with Digital Twins

Mountain Connect 2021

Rob Tiffany

Ericsson

8-10-2020

Hi, I’m Rob

Vice President and Head of IoT Strategy at Ericsson and currently developing the “Thunderstruck” system to accelerate Digital Transformation

Executive Director at the Moab Foundation

Board Director at Elevate Our Kids

CTO and creator of the Lumada Industrial IoT platform and Asset Avatar digital twins at Hitachi

Global Technology Lead for the Internet of Things at Microsoft

Session Objectives and Takeaways

Session Objectives:

Discover the history of Digital Twins and why we use them

Learn about Digital Twin Models and associated Properties

Apply Rules and KPIs to Digital Twin Instances

Learn advanced Digital Twin topics

Key Takeaways:

Be able to describe the value of Digital Twins and the basics of how they operate

What are they?

A digital twin is a digital representation of a physical object, process, place or system and sits at the intersection of Connectivity, IoT and AI.

Digital twins are used in both design, simulation, and operational phases of a product's lifecycle.

Products are designed digitally before physical creation

Digital simulations are performed by feeding test data to see how digital twins react.

Operationally, IoT data populates the digital twin with the physical counterpart's current state and behavior.

Software bots compare captured data with expected state values and KPIs to trigger further analysis and actions.

Where did Digital Twins Come From?

NASA used a physical twin in Houston to get Apollo 13 home.

The concept of digital twins were anticipated by David Gelernter's 1991 book “Mirror Worlds.”

Dr. Michael Grieves introduced the “Doubleganger” (digital twin) as part of product lifecycle management (PLM) at the University of Michigan in 2002.

Dr. Grieves went on to say, “Industry 4.0 is only possible with the digital twin.”

Digital Twin Models

A digital twin model is used to define a type of entity or thing.

It includes a baseline of information including a name, description, version number and a picture or CAD model.

Allows you define all aspects of a type of thing just once, rather than defining it repeatedly for each individual thing in your IoT platform.

Each instance of a digital twin derived from a digital twin model will inherit its properties.

Telemetry Properties

Telemetry is IoT data coming from sensors.

The telemetry properties of a digital twin model tells the event processing engine and bots in your Internet of Things platform what to expect.

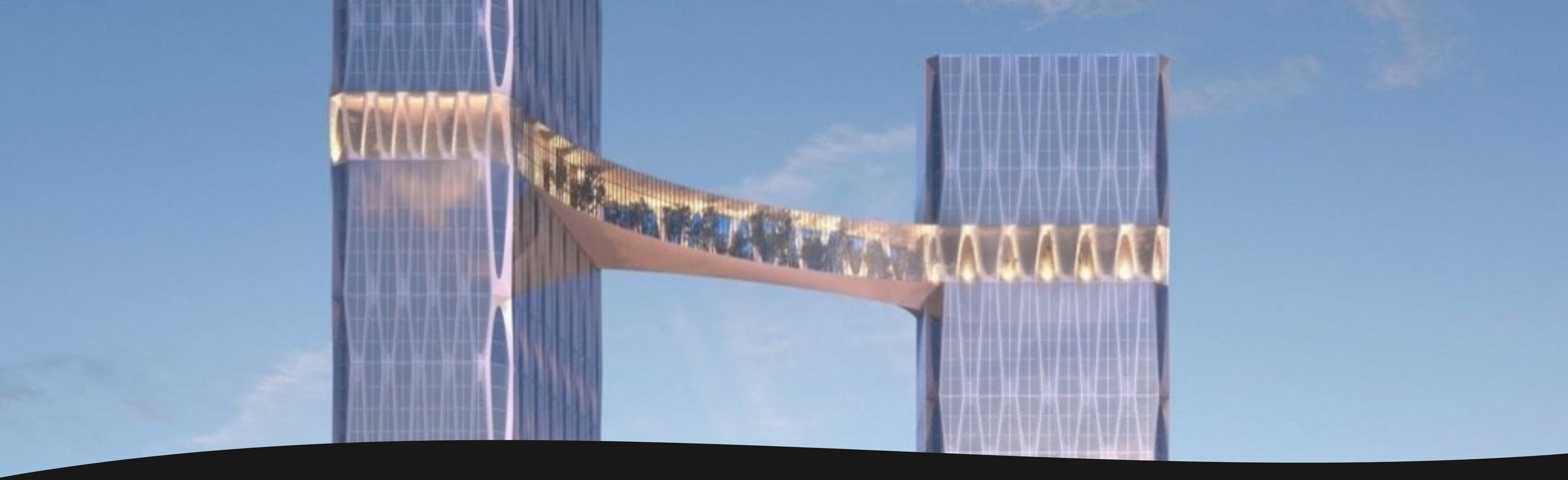
This includes data labels, data types, and units of measure for the incoming data to assist with pattern matching.

Virtual Properties

A calculated property that doesn’t have a 1:1 relationship with the sensors that are actually sending data from a device.

The value assigned to a virtual property is typically derived from a mathematical combination of values from one or more telemetry properties and possibly other reference data.

For example, calculating miles per hour (speed) of a car is derived from a combination of telemetry properties like the rotating drive shaft and magnetic sensors using simple analytics to tell you how fast you’re going.



Static Properties

Properties that contain values that don’t typically change.

Using a car as an example, static properties could be things like the length of the car, the number of cylinders and displacement of the engine as well as the volume of room in the trunk.

Static properties are necessary to have a more complete view of the actual entity and to be used as reference data for analytics.

Command Properties

If your IoT platform needs to work with industrial control systems where it's necessary to send messages to trigger electric, hydraulic, pneumatic, or mechanical actuators, you'll define one or more command properties.

They include parameters such as names, data types, and units of measurement to assist a command in properly triggering the actuator.

For example, an electric motor lets you remotely set its revolutions per minute (RPM) to control the speed. You define the name of this actuation command "speed" as specified by the manufacturer of the motor. You'll need the data type "integer" of the value to send. Lastly, you can guess the unit of measurement is "RPM."

Process Properties

A process is a series of actions, tasks or steps taken in a linear or sometimes a branching, non-linear sequence in order to achieve a desired outcome.

These process steps could be manual activities undertaken by a person, purely digital steps taken by software across computers, electromechanical actions between digital messages and mechanical actuators as well as cyber-physical tasks performed by industrial robots.

Steps needed to guide software bots in taking actions to achieve desired outcomes are defined. Each step is represented by a process property. Each process property defines the digital twins involved, the APIs needed to connect, security requirements for calling APIs, data to be sent, return values to expect and other details needed to successfully complete the step and move on to the next one.

The Digital Twin Instance

When you entangle a physical entity with a matching digital twin model and associated properties, you derive a digital twin instance.

The process starts by defining a unique identity that binds the physical entity with the digital twin in the IoT platform.

Authentication is facilitated through the use of security tokens, API keys or X.509 certificates before data is allowed to enter the IoT platform to populate the digital twin instances.

The current and historical state of every individual + fleets of twins becomes available for analytics.

Basic Rules

Deriving value from streaming telemetry data revolves around pattern matching, key performance indicators (KPIs) and filtering by specifying one or more rules to be associated with each telemetry property.

This is accomplished using simple operators such as *equals*, *not equals*, *greater than*, *greater than or equal to*, *less than*, & *less than or equal to*.

Define a telemetry property for the "left front tire pressure" of a car with an "integer" data type and "PSI" unit of measure. To create a "green" KPI between 30 & 35 PSI, you define a rule where bots look for values that are >= 30 and <= 35.

Using simple IFTTT algorithms, the event processing engine of your Edge or Core IoT platform would apply those rules to incoming data and bots trigger an action for values outside that range.

Subsystems

Most IoT platforms have you define a simple data schema for a machine which is fine unless the entity you're monitoring is comprised of multiple, complex subsystems.

For example, a car is a system made up of many subsystems including the engine, braking, transmission, electrical and fuel subsystems. Some of those subsystems deserve to be digital twins with telemetry, virtual, static & command properties of their own.

These subsystem digital twins have a parent/child relationship with the overall car and causal relationships with each other's properties.

If the engine doesn't run when you start your car, the cause could be the battery, starter or alternator. Defined causal relationships between the engine and properties of the electrical subsystem would alert you to the correct cause. This helps you get prescriptive analytics.

Groups

Physical entities don’t live in a vacuum, they operate in larger systems of systems with relationships and interactions with other entities.

Creating a group called “assembly line” to represent a collection of industrial robots that work together does them a disservice.

The assembly line should be a digital twin itself where telemetry, virtual, static, and command properties have defined causal, parent/child relationships with all the other robots and the robots have peer relationships with each other.

Collections of assembly line digital twins can then come together to create a composite digital twin called a factory.

The Digital Thread

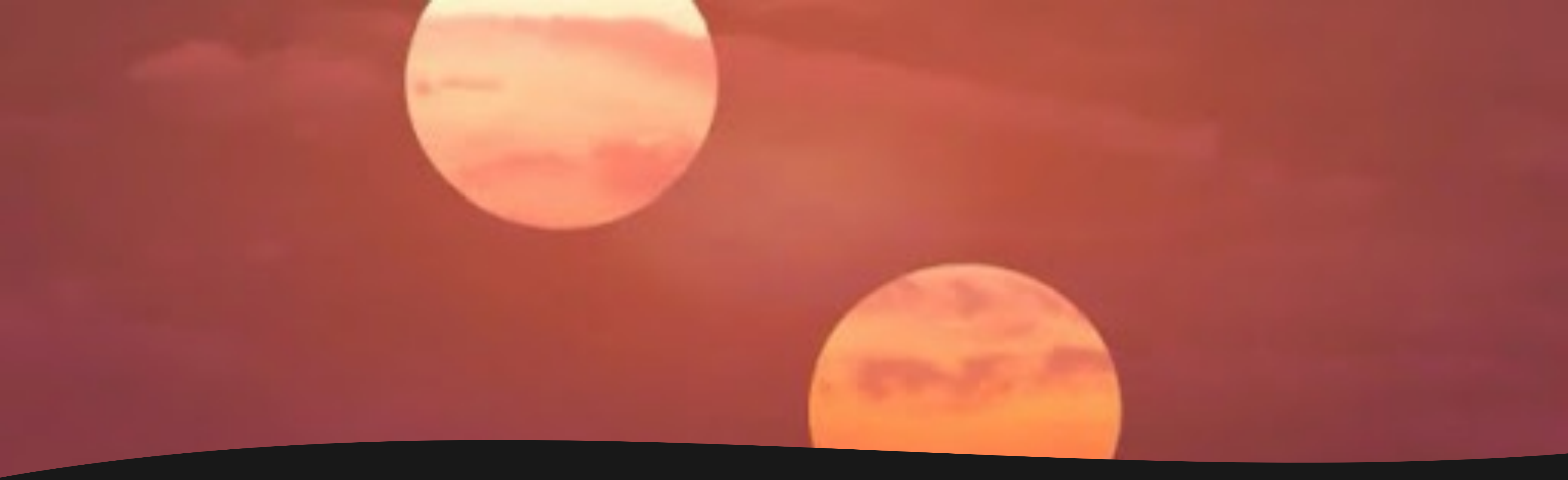
A historical record of what happens to an instance of a Digital Twin throughout its entire lifecycle, from design to decommission, is represented by something called a Digital Thread.

Beyond the IoT telemetry the digital twin captures from the physical entity, other significant events are captured via an ever-growing digital thread.

Taking a car to the shop for an oil change, repair or performing routine service all represent events that are manually added to the digital thread.

Imagine capturing an event where a car suffered a water pump failure at 60,000 miles. Sharing this information with the digital twins of other cars of the same type would be invaluable.

The digital thread tells you the story of the machine.



Summary

To summarize, physical entities equipped with compute, storage, and networking capabilities feed IoT telemetry data into digital twins with the help of an IoT platform.

The digital twins are inherently capable of performing ***descriptive analytics*** to tell you "what has happened" and "what is happening."

With the help of bots, KPIs, statistical models, and advanced analytics, digital twins can deliver ***predictive analytics*** to tell you "what is going to happen?"

Lastly, digital twins can define an array of ***prescriptive analytic*** outcomes that tell you "what you should do" for a given descriptive or predictive insight.


Ask me some Questions