



*WHITEPAPER*

## Digital Twin Technology

How to develop real-time models, synchronized with life data and augmented with intelligence.

# Introduction

## What is a Digital Twin?

In short, a digital twin is a virtual version of something real, where the real thing and its virtual version are continuously synchronized. The 'real thing' can be anything, like a machine, a river, a greenhouse, a human heart or a logistical process.

In a basic form, a digital twin is an intuitive monitoring and control application. But real benefits come from simulation models, analysis tools and forecasting capabilities in the digital twin that allow exploration of what-if scenarios, advanced forms of control and optimization, operator training and troubleshooting.

Apart from their immediate use, digital twins also naturally collect large, semantically rich and labeled data sets that can be used to train AI applications. Incorporating such AI applications in the digital twin makes them even more intelligent and useful.

## Digital Twin Technology

Digital twins have significant benefits. But developing a good digital twin is far from trivial. Simulation models and forecasting algorithms in the digital twin need to be extremely fast as they must work in real time. The models need to be continuously synchronized with incoming sensor data that is typically unreliable and noisy. And any built-in intelligence should be reliable enough for users to trust the digital twin.

All this requires advanced technology. This technology is available today even if it has partly only just emerged from universities. This whitepaper gives an introduction to digital twins and provides an overview of the underlying digital twin technology. It is intended for companies that consider developing a digital twin but lack the knowledge or the capacity for certain aspects.

Before we dive in, a word of caution: the technology that is discussed here develops fast. Therefore, this whitepaper may already be outdated when you read it. If you want to get in touch with the latest developments, feel free to contact us. See [www.vortech.nl/contact](http://www.vortech.nl/contact) for our contact details.

# Digital twins: a definition

Many definitions of the digital twin concept exist. For this whitepaper, we'll use one from Juniper Research, which is a very complete and appropriate one:

**A digital twin is a virtual model representation of a connected physical product, process, or service, across its whole lifecycle (design, build, operate). The virtual replica uses operational real-time data and other sources of information to enable detection of issues, advance both learning and understanding, as well as test and simulate scenarios in the physical model counterpart.**

What makes this definition so wonderful is that it nicely sums up all the relevant aspects of digital twins:

- **The digital twin is connected to the real thing**, which means that there is mostly continuous communication between the digital twin and its real counterpart, ideally two-way.
- **A digital twin can represent an actual product, but also a process, like the various steps in assembling a device, or a service like the delivery of parcels.** Also, the term product should be understood widely and can apply to anything: it can also be a river, a human heart or a greenhouse.
- **A digital twin ideally exists across the whole lifetime of its real-life counterpart.** Consider for example a turbine. The digital trace of the turbine starts already with digital design documentation and probably also the results of design simulations. During construction, commissioning and operation, all sorts of digital information are added like test results of parts, logs etc. When decommissioning the turbine, a digital twin will be useful to know what materials are in there, but the decommissioning will itself also generate digital data, like where the discarded parts have gone.
- **Digital twins not only use real-time data but also other sources of information.** As we just saw, the design documentation is typically also part of the digital twin of a turbine as it may contain information that can be important during operation. Likewise, when talking about the digital twin of a human heart, a lot of context information is important like the diet of the owner of the heart, her age and any disease history.

- **An important use of digital twins is to detect issues.** For example, the temperature in a greenhouse may rise unexpectedly due to a failing sunscreen. The digital twin will notice that the temperature goes out of bound and may be intelligent enough to ascribe this to the failing sunscreen.
- **Another benefit of digital twins is their use for learning and analysis.** This is reinforced by the fact that an offline copy of the digital twin can be used to experiment with settings and control strategies without jeopardizing the real thing. This also leads to better understanding.

# Classifications of digital twins

Although the definition of a digital twin as given above is the full version, not all digital twins implement the full functionality. Therefore, it's useful to discern different classes of digital twins. An extensive discussion of this topic can be found in [this blog post](#) but we'll briefly mention two classifications to deepen the understanding of the digital twins concept.

## Simulations, shadows and twins

In practice, people use the term digital twin for things that are not formally digital twins:

- A traditional, offline **simulation** of something is also a sort of virtual copy but does not qualify as a digital twin as it lacks the essential aspect of real time synchronization.
- When the virtual copy is continuously updated by data from the real-world twin but there is no data going the other way, it's called a **digital shadow**.
- The full digital twin concept requires that the virtual copy is continuously synchronized with the real-world twin and, in return, can send control signals to the real-world twin.

## Single, multiple or composite instances

- A digital twin may refer to a **product category**, like a certain type of turbine. It may be useful to collect data from all real-life instances of the turbine within the digital twin of this product category, for example to learn about operational variations. The communication from the digital twin back to the real-life instances can be meaningful for example for software updates.
- A digital twin may refer to a **single instance** of a product category, like one specific turbine in a specific company.
- A digital twin may refer to a **composite** of digital twins, for example when digital twins for a number of different turbines are combined into a digital twin of a powerstation.

# Elements of a digital twin

Conceptually, digital twins consist of three parts: the data store, the user interface and the intelligence.

## The data store

The data store holds all the information about the real-life twin. This includes not only the life data stream coming in from the real-life twin, but also all sorts of meta-information such as designs, commissioning information, and the bill of material. In addition, the data store may also hold analysis results and simulation results that are produced by operators doing what-if scenarios.

## An intuitive user interface

This is the most visible aspect of the digital twin for its users. When the real-life twin is an object, the user interface typically displays an interactive 3D rendering of the object with the option to click on the rendering to obtain specific information. For digital twins of processes or services, the interface is usually in terms of schematics. Through the user interface, the user can interact with the digital twin for simulation and analysis. Optionally, the user can indirectly control the real-life twin by forwarding operations on the digital twin to its real-life counterpart.

## The intelligence component

This includes simulation models, analysis applications and AI-tools.

- For many digital twins, the simulation model is its core: it runs continuously and is fully synchronized with the real-life twin. In this synchronized simulation, the user can also inspect quantities that are not actually observed but are computed by the simulation (what is known as **soft sensing**). This is one of the many reasons why digital twins are so powerful. Apart from their real-time application, simulation models are also used for what-if exploration.
- Analysis tools offer functionality to help the user better understand what is going on. These may be very simple tools, like averaging a signal over time, but also more advanced tools like outlier detection and functionality for predictive maintenance.
- The category of AI-tools is still very much in development. The trend is towards tools that learn to understand the real-world twin and offer the user advice on how to control the real-life twin. AI-models can also be used as a stand-in for physics-based simulations (what is called **Physics AI**).

# Digital twin technology

Apart from the usual IT-aspects like proper interconnects and databases, there are a number of technologies that are specific to digital twins. We'll discuss them in this chapter.

## Fast simulation and computing

The core of a digital twin is the computational model or simulation that is continuously synchronized with the real-life twin. That means that the model must also run real-time. Or even faster when doing what-if scenarios or model-in-the-loop control. At the same time, the model needs to be accurate and reliable. All this puts strong demands on computing speed.

An obvious solution is to use powerful hardware, but that will only get you so far. In most cases, the model itself must be optimized. There are several approaches to this (see also our [whitepaper](#) on speeding up computations). The computing code can be optimized to make the best use of the hardware. For example, it can be tailored to use [GPUs](#). Alternatively, models can be replaced by surrogate models. These can either be developed through [Reduced Order Modelling](#) approaches or through [Physics AI](#) (the combination of machine learning with first-principle scientific insights).

## Continuous synchronization

The computational model or simulation must be constantly synchronized with the real-life twin. This not only means that all control signals that are applied to the real-life twin must also be applied to the digital twin (or the other way around), but it also means that all sensor readings from the real-life twin must be used to continuously calibrate the digital twin. For example, a turbine may experience wear, leading to a higher resistance which will show up in the temperature sensors. These temperature readings can inform an update on the wear-parameter.

Using sensor data to update a model of the real-life twin is known as [data-assimilation](#). This comes in two basic flavors: automatic calibration and filtering. Passing over many nuances, it can be said that with automatic calibration, the **model parameters** are automatically set such that the model results match the sensor observations. With filtering, the **model state** is continuously updated to match the sensor readings while still maintaining a physically correct state.

Many algorithms and open-source tools are available for this synchronization.

## Machine learning

Machine learning is the key technology underlying much of the intelligence part of a digital twin. This term encompasses a wide range of algorithms and approaches. Today, Large Language Models such as ChatGPT are well-known. As language models, they can be useful in a digital twin to support the user: they may suggest actions to take in certain situations, or they may answer questions about the real-life digital twin based on the data in the database. In the form of an agent, an LLM could take actions all by itself, but this functionality is rarely enabled or even implemented as agents are often not yet sufficiently trusted.

A different type of machine learning works on the physical side of things: it learns to understand the physical behavior of the real-life twin and to predict the effects of control actions or the progress of degradation. This type of machine learning is called [Physics AI](#) and uses a combination of sensor data and scientific knowledge to create models that are scientifically sound and match the available data. The methods that are used for Physics AI are mostly different from those used for LLMs, although basic concepts like neural networks underly both.

## Advanced visualization

The graphical part of a digital twin is an important aspect as it contributes much to making the digital twin intuitive and interactive. The 3D-representation of the twin is not too special and is usually built on one of the graphics libraries like those that are used for gaming.

Where it gets more interesting is in the visualization of the data. Getting the displays right is as important as it is specific: users typically know best what kind of information is useful for them and what options for interaction they need. Therefore, this is best done in close [collaboration](#) with the users and in an iterative way until the result is just right for the user. Apart from this interplay between user and developer, good visualization also requires skills to process large quantities of data quickly and correctly.



# The business case for digital twins

Implementing and maintaining a digital twin costs money. Therefore, the benefits should be clear. A full discussion of this topic can be found in [this blog](#). Here we briefly discuss the most important use cases, where the introduction of artificial intelligence is by far the most important.

## Advanced Monitoring and control

On the most basic level, the digital twin is no more than a SCADA system. You want to monitor and control your physical assets and also collect data about them for future use. Such future use could be simply to know what happened at a certain moment. Or it could be the analysis of the historical data to find optimizations. Again, these types of digital twins are little more than what we already have had for decades and their value is the same as ever.

## Running what-if scenarios

Most people would not call a SCADA system a digital twin, even if it has a nice 3D graphics interface. The added value of digital twins really comes from the functionality to simulate and run what-if scenarios. This means that you can do virtual experiments with operational settings of an asset or process without the risk to damage anything. It's like a computer game where you can just restart after being shot down (or have any other fatal event happen to you). If the simulation shows that your machines will be harmed, you just try something else. Right until you've found the sweet spot where your operation is optimal and nothing gets broken.

## Operator training

The simulation aspect of digital twins is also an excellent tool to train new operators. This, in itself, is a significant advantage as there is a certain worry in several sectors that experienced operators will soon retire and new ones with the right qualifications are not readily available. Having a good training facility for new operators can therefore prove to be very important.

These days, digital twins are also starting to get used to train robots or machine learning models that may someday replace part of the operator's tasks.

## Introducing artificial intelligence

Artificial intelligence promises great leaps in productivity, but it needs some digital representation of reality to work on. Digital twins offer this digital representation and form a basis for artificial intelligence or agents to do what they are trained to do. At the same time, the actions of the artificial intelligence or agent can be restricted to the digital twin and not be passed on to the real-life twin until a human operator is comfortable to do so. At least until artificial intelligence and agents become fully reliable, this human-in-the-loop approach prevents serious accidents.





# VORtech: expert in digital twin technology

VORtech has been helping its clients to develop software for simulation and computing for more than 30 years. Many of the applications that we worked on were precursors to what is now called a digital twin.

Our work has taught us how to speed up computations, how to do data-assimilation, how to apply machine learning and how to build solid applications with all of that. This has given us a portfolio of knowledge and experience that is perfect for supporting our clients in developing digital twins.

## Collaboration skills

Our developers typically come from an engineering background, which allows them to collaborate seamlessly with domain experts in most technical or scientific sectors. In addition to this engineering savvy, they are specialized in mathematics and algorithmics, which lets them select and implement the most appropriate computational approaches for any given challenge. Finally, they've learned to build software that is both efficient and maintainable.

## Domain agnostic

VORtech works in many different sectors. We are obviously not a domain expert in all those sectors, but our ability to collaborate easily with domain experts allows us to make useful contributions every time. What we bring to the table is our knowledge of advanced computing and simulation whereas the client's experts provide the necessary domain knowledge. Together, we can build the best tools for any industry.

## A partner for our clients

We position ourselves as partners for our clients. This means, to us, that we will shape our activities to best match the needs of the client. Sometimes, a client may need a specific tool to be implemented, sometimes they want to explore new technologies with us. Sometimes they just need extra development capacity and sometimes they need an experienced eye to audit their software. All this can be accommodated.

## Get to know us

If you are interested in our support for the development of a digital twin, feel free to contact us at [www.vortech.nl/contact](http://www.vortech.nl/contact). We'll be happy to come and discuss your needs with you and see if we can support you.