



Digital Twins IN ACTION

Greg Biegel

Digital Twins in Action

Digital Twins in Action

GREG BIEGEL



MANNING
SHELTER ISLAND

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

©2026 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

- ⊗ Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

The author and publisher have made every effort to ensure that the information in this book was correct at press time. The author and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause, or from any usage of the information herein.

 Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Development editor: Rebecca Senninger
Technical editor: Stephen Witherden
Review editor: Angelina Lazukić
Production editor: Kathy Rossland
Copy editor: Andy Carroll
Proofreader: Melody Dolab
Typesetter and cover designer: Marija Tudor

ISBN 9781633434653
Printed in the United States of America

contents

<i>preface</i>	<i>xii</i>
<i>acknowledgments</i>	<i>xiv</i>
<i>about this book</i>	<i>xvi</i>
<i>about the author</i>	<i>xix</i>
<i>about the cover illustration</i>	<i>xx</i>

1 *Bridging the physical and digital worlds* 1

1.1	What is a digital twin?	2
1.2	Technologies enabling digital twins	3
	<i>Internet of things</i>	<i>4</i>
	<i>Cloud computing</i>	<i>5</i>
	<i>Machine learning and artificial intelligence</i>	<i>6</i>
	<i>Visualization technologies and game engines</i>	<i>7</i>
	<i>Agentic AI and autonomous decision-making</i>	<i>7</i>
1.3	Digital twin capability levels	8
	<i>Descriptive digital twin</i>	<i>9</i>
	<i>Informative digital twin</i>	<i>11</i>
	<i>Predictive digital twin</i>	<i>12</i>
	<i>Comprehensive digital twin</i>	<i>15</i>
	<i>Autonomous digital twin</i>	<i>16</i>
1.4	What are digital twins used for?	18
	<i>Accelerating product development</i>	<i>18</i>
	<i>Reducing costs through predictive maintenance</i>	<i>19</i>
	<i>Optimizing performance and operational efficiency</i>	<i>19</i>
	<i>Training and simulation</i>	<i>19</i>
1.5	Digital twins across industries	20
	<i>Mining, energy, and industrial</i>	<i>20</i>
	<i>Automotive</i>	<i>20</i>
	<i>Agriculture</i>	<i>20</i>
	<i>Defense</i>	<i>21</i>
	<i>Healthcare</i>	<i>21</i>

Infrastructure 21 ■ Smart buildings, cities, and states 21

What a digital twin is not good for 22

1.6 Building your first digital twin: Key considerations and challenges 22

A clear definition of success 24 ■ Data quality 24 ■ Skills gap 25

Build vs. buy 25 ■ Governance and security 26

2 Mapping physical systems to digital representations 28

2.1 Defining your objectives 29

Identify specific challenges and opportunities 29 ■ Establish measurable performance indicators and baselines 29

Map decision-making improvements 30 ■ Define a minimum viable digital representation 30

2.2 A home digital twin to optimize energy and water use 31

Mapping the home to a digital representation 32 ■ Start with what you have 33

2.3 Information sources for digital representation 33

Historical records 33 ■ Photographs 36 ■ Videos 37

Engineering documents 40 ■ External systems 43

2.4 Spatial and geometric representations 43

2D geometric models 44 ■ 3D geometric models 46

2.5 Spatial mapping and reference systems 50

Coordinate systems and projections 50 ■ Coordinate transformations and conversions 52

2.6 Deciding what you need 55

3 Sensing the real world 58

3.1 How sensors work 59

The sensor signal chain 59 ■ Key sensor characteristics 61

3.2 Selecting the right sensors for your digital twin 61

Defining what you're measuring 62 ■ Matching sensor capabilities to real-world constraints 62 ■ Verifying integration before purchasing 64 ■ Choosing sensors for a home digital twin 65

3.3 Connecting sensors: Communication technologies 66

Short-range technologies (<100 m) 67 ■ Long-range technologies (>100 m) 68 ■ Choosing the right technology 68

Universal data transport architecture 70

- 3.4 Real-world data collection strategies 72
 - Edge processing* 72 ▪ *When electronic sensing isn't enough* 73
 - Manual sensing strategies* 74
- 3.5 Data processing 76
 - Understanding IoT data characteristics* 76 ▪ *Message decoding* 77 ▪ *Sensor fusion* 79
- 3.6 Scalability and device management 81
- 3.7 Bringing it all together 83
 - Data flow walkthrough* 83 ▪ *Architectural principles* 84

4 Data integration and management 86

- 4.1 Types of data 87
 - Reference data* 87 ▪ *Time-series data* 87 ▪ *Unstructured and semi-structured data* 88 ▪ *Spatial data* 89
 - Derived data* 90
- 4.2 Data sources 91
 - OT data sources* 92 ▪ *IT data sources* 93 ▪ *The convergence of IT and OT* 94 ▪ *External data sources* 95
- 4.3 Data structures and storage 96
 - Relational and transactional data* 96 ▪ *Columnar and analytical storage* 98 ▪ *Time-series data storage* 99
 - Semi-structured and document data* 99 ▪ *Object storage, data lakes, and lakehouses* 100 ▪ *Specialized storage systems* 102 ▪ *Data life cycle management* 102
- 4.4 Data ingestion 103
 - Batch data ingestion* 103 ▪ *Streaming data ingestion* 104 ▪ *API integration* 106 ▪ *ETL and data transformation* 107
- 4.5 Data governance and compliance 109
 - Data classification and sensitivity* 109 ▪ *Regulatory compliance* 109 ▪ *Access control and authentication* 109
 - Data quality* 110
- 4.6 Selecting the right combination 111
- 4.7 Data store for a home digital twin 112
 - Storing time-series data* 112 ▪ *Storing object data* 113
 - Storing relationship data* 114

5 *Modeling reality* 116

- 5.1 Making sense of data 117
 - From measurements to decisions* 118 ▪ *The knowledge engineering challenge* 119
- 5.2 Understanding context in digital twins 119
 - Types of context* 121 ▪ *How to contextualize data* 122
 - The importance of context in generative AI* 124
- 5.3 Ontologies as a blueprint for understanding 125
 - Core components of an ontology for a digital twin* 125
 - Defining ontologies for digital twins* 125 ▪ *Choosing an ontology for your model* 126
- 5.4 Knowledge graphs 128
 - Graph theory* 128 ▪ *Building a knowledge graph as a labeled property graph* 129 ▪ *Graph traversals* 130 ▪ *Building a knowledge graph of my home* 132 ▪ *Choosing a knowledge graph service* 133
- 5.5 Standards and interoperability 134
 - A framework for implementing standards in your digital twin* 134
 - Practical standards assessment* 134 ▪ *A practical assessment framework* 135 ▪ *Real-world example: The Asset Administration Shell* 135
- 5.6 Serving the model of reality 137
 - Context-aware data retrieval* 137 ▪ *Building an API layer for my home digital twin* 139 ▪ *The power of a GraphQL schema* 140

6 *2D Visualization and insight* 144

- 6.1 Why graphical visualization matters in digital twins 145
 - Data-to-visual mapping principles* 145 ▪ *User experience considerations* 147
- 6.2 2D visualization techniques 148
 - Charts, graphs, and data visualization* 148 ▪ *Dashboards and control panels* 152 ▪ *Schematic diagrams* 155 ▪ *Spatial 2D visualization* 157 ▪ *Heatmaps* 159
- 6.3 2D visualization performance considerations 159
 - Choosing the right rendering technology* 160 ▪ *Data decimation and aggregation for time-series data* 161
- 6.4 Building effective visualizations with Grafana 163
 - Why Grafana works well for digital twins* 164 ▪ *Choosing a visualization platform* 164

7 *Spatial context and 3D insight* 170

- 7.1 3D visualization fundamentals for digital twins 171
 - More complex models* 173 ▪ *Texturing the mesh* 174
 - Web-based 3D visualization* 176
- 7.2 Creating a mesh-based 3D geometric model 176
 - Serving the model via the web* 177 ▪ *Spatially locating the model* 179
- 7.3 Data integration 183
 - The importance of spatially locating data in 3D* 183
 - Annotation and markup* 183 ▪ *3D visualization performance considerations* 185
- 7.4 Capturing reality in 3D 186
 - 360-degree photography* 187 ▪ *Photogrammetry* 187
 - Neural Radiance Fields* 189 ▪ *Gaussian splatting* 190
 - Light detection and ranging* 191 ▪ *Simultaneous localization and mapping* 192 ▪ *Universal Scene Description and the NVIDIA Omniverse* 193
- 7.5 Extended reality 195
 - Virtual reality: Full immersion* 195 ▪ *Augmented reality: Contextual overlay* 197 ▪ *Mixed reality: Blending worlds* 198
- 7.6 Building effective visualizations 198
 - Security and access control* 200 ▪ *Usability and user experience* 200 ▪ *Data-to-visualization pipeline and scalability* 200

8 *Integrating inference and intelligence* 203

- 8.1 A practical data pipeline 204
 - Cleaning and normalization* 205 ▪ *Time alignment and resampling* 206 ▪ *Feature extraction* 207 ▪ *Storage* 210
- 8.2 Anomaly and fault detection 210
 - Rule-based detection and its limitations* 211 ▪ *Statistical approaches* 211 ▪ *Unsupervised learning* 213
- 8.3 Predictive modeling and machine learning 215
 - Training a forecasting model* 215 ▪ *Applying AutoML* 217
 - Supervised learning* 219
- 8.4 Generative AI 221
 - Retrieval-augmented generation* 222 ▪ *Agentic AI* 224
 - Building an agent for the home digital twin* 224 ▪ *When agents*

complement or replace control strategies 228 ▪ *Multi-agent systems* 228

8.5 Choosing the right approach 229

Operational viability 230 ▪ *Safety and human in the loop* 230
Cost 231

9 *Predicting outcomes with simulation* 233

9.1 Modeling and simulating physical systems 234

The digital twin advantage 234 ▪ *From static models to dynamic simulation* 234 ▪ *Continuous versus discrete event simulation* 235

9.2 Continuous systems 236

Simulating a continuous system 237 ▪ *The initial value problem* 237

9.3 Acausal modeling for digital twins 239

How it works 240 ▪ *Modeling tank drainage in OpenModelica* 240 ▪ *Executing simulations* 243 ▪ *Integrating sensor data with continuous simulation* 245

9.4 Simulating processes with discrete event simulation 248

Why DES matters for digital twins 248 ▪ *DES with Python using SimPy* 248 ▪ *Example application of DES in digital twins* 251

9.5 Finite element and field simulation 252

Using FEM for virtual sensing in a digital twin 254 ▪ *Example application of FEA in digital twins* 255

9.6 Computational fluid dynamics 256

CFD with Python for lightweight simulation 256 ▪ *Example application of CFD in digital twins* 259

9.7 Reduced-order models 259

ROMs for virtual sensing 259 ▪ *An example ROM* 259

10 *Digital twins in production* 263

10.1 Selecting the right deployment platform 264

Public cloud providers 264 ▪ *Digital twin development platforms and engines* 265 ▪ *Fully managed digital twin platforms* 266

10.2 Security as a cross-cutting concern 266

Expanding the threat model 267 ▪ *Identity everywhere* 268
Policy as code 269

10.3	Digital twin operations	270
	<i>Observability</i>	270
	▪ <i>Model lifecycle management</i>	272
	▪ <i>Change management</i>	274
	▪ <i>Cost management</i>	275
	▪ <i>Software dependency management</i>	276
10.4	Aligning to strategy and measuring value	277
	<i>Operational impact</i>	277
	▪ <i>User adoption and trust</i>	278
	<i>Financial value</i>	278
10.5	Governance, ethics, and responsibility	279
	<i>Decision authority</i>	279
	▪ <i>Accountability and ownership</i>	280
	<i>Ethical use</i>	281
	▪ <i>Auditability and traceability</i>	281
10.6	A production readiness checklist	283
<i>appendix A</i>	<i>Building a LoRaWAN network</i>	287
<i>appendix B</i>	<i>Building a custom IoT sensor</i>	296
<i>appendix C</i>	<i>Capturing a 3D model using photogrammetry</i>	305
	<i>index</i>	313

preface

The term “digital twin” has been applied to many different kinds of systems. From its origins in product lifecycle management in the early 2000s to today’s sophisticated, photorealistic, and physically accurate replicas powered by GPU-accelerated simulation and AI, the label has been applied to many different types of systems. Despite this, the core building blocks remain the same—you create a digital representation of a physical system, keep it synchronized with real world changes, and use it to make better decisions. This book describes the journey you will follow when you decide that building a digital twin will deliver real value.

As AI increasingly shapes our world, digital twins have become more important than ever. They provide the structured, up-to-date context grounded in reality that our models need to reason about real-world systems. Without that grounding, even the most advanced AI will struggle to move from generic outputs to meaningful, actionable insights.

Writing about digital twins is challenging because they are not a single technology but a convergence of many. Modeling, data storage, integration, internet of things (IoT), simulation, machine learning, visualization, and distributed systems all play a role, and each of these topics could easily fill a book on its own. In my experience helping organizations build digital twins, I have found that much of the available guidance was either too high-level to be actionable or too academic to be practical. What was missing was a clear, hands-on path for practitioners.

Certain challenges arise repeatedly when building digital twins. Teams struggle to extract information from documents, images, videos, and sensors; they wrestle with connecting fragmented data across siloed systems; and they fight to turn that data into usable insights for both people and software agents. Overcoming these hurdles is the core focus of this book.

To make these abstract concepts concrete, I thread a running example through the chapters: building a digital twin of a home. I chose this example deliberately. Everyone has a home, which means the domain requires no specialized knowledge to follow. The techniques involved—including ingesting heterogeneous data, building a spatial representation, synchronizing state, running simulations, and applying ML and AI—are the same techniques you would use to build a digital twin of a factory floor, a power grid, or a fleet of vehicles. The home example simply lets me focus on method rather than domain complexity.

That said, industrial digital twins introduce concerns that a home example cannot fully exercise. Governance, security, autonomous decision-making, regulatory compliance, multistakeholder coordination, and organizational change management all become significantly more complex at industrial scale. I address these topics where they arise naturally in the text and point to dedicated resources where a deeper treatment is warranted. My goal is to give you a foundation that transfers, while being honest about where the additional complexity lies.

When you need to go deeper, there is a wealth of excellent information available on each of the underlying technologies. This book is intended to give you the foundation and the mental model and practical approach you will need to begin your own digital twin journey with confidence.

acknowledgments

Writing a book is a demanding process, and I am deeply grateful to my family for making it possible. My wife, Hannah, took on so much to give me the space I needed to write, and my daughters, Lucinda and Annika, were incredibly forgiving of the hours I spent at my keyboard instead of with them. Thank you all.

I owe a great deal to my editor, Rebecca Senninger, and would like to thank her for her expert guidance, advice, and insightful feedback throughout the process. Rebecca is a fantastic editor who kept me on track and helped shape the book into what it is today.

I was incredibly fortunate to have Stephen Witherden as my technical reviewer. Stephen's attention to detail and practical perspective elevated every chapter, dramatically improving the book's clarity and accuracy. Stephen Witherden leads software engineering for Beca, a 4,000+ strong engineering professional services consultancy in the Asia Pacific. As technical editor for this book, he brought his internationally recognized practical experience in delivering digital twins for organizations across defense, water, manufacturing, primary industries, and buildings.

I would also like to thank the wonderful team at Manning Publications, including Michael Stephens, for their professionalism and support throughout the journey from idea to finished manuscript.

My thanks go to the additional technical reviewers—Alex Mendes da Costa, Alexey Agarkov, Alvin Yap, Amit Chokshi, Anand Stephen, Anna Ptaszniak, Aparna Achanta, Bireswar Banerjee, Bulent Soykan, Detlev Richter, Dima Timofeev, Dipen Pradhan, Michael Grieves, Frederik (Freek) Boersma, Hilmar Retief, Hubert Homaei, Ishmael Diaw, Izazahmad H. Doddamani, Jean-Paul Malherbe, Jenny Vega, Jigar Kothari, John Holmes, John McCormack, Lesia Yanytska, Manish Rungta, Marco Massenzio, Mary

Olive Atim, Maulik Bhat, Mike Williams, Navin Manaswi, Nikhil Kassetty, Olaf Burau, Pierpaolo Vergati, Ricardo Bánffy, Rob Foster, Roberto Casadei, Sanjay Basu, Stefano Fago, Steve Grey-Wilson, Steven Fernandez, and William LeBorgne—who took the time to read early drafts and provide valuable feedback. Their insights helped refine both the explanations and the examples, ensuring the content remained grounded in real-world practice.

Finally, thank you to the colleagues, mentors, and leaders I have learned from throughout my career. The systems we’ve built, the problems we’ve solved, and the conversations we’ve shared are the true foundation of this book.

about this book

Digital Twins in Action aims to show you how to build a digital representation of a physical system that you will use to make better decisions. A better decision in this context simply means achieving an outcome faster, more safely, with fewer resources, or a combination of all these.

The book uses a home digital twin as its running example, not because home automation is the target audience, but because the underlying techniques transfer directly to industrial, infrastructure, and enterprise systems. The home provides a universally accessible context for learning the core skills involved in building a digital twin: translating a physical system into a digital representation meaningful to humans and machines, using a variety of data to keep that model synchronized with reality, building visualizations of the digital model, and using it to predict future outcomes through simulation, ML, and AI.

Who should read this book

This book is for anyone interested in understanding what is involved in building a digital twin. It is aimed at practitioners who want to move beyond theory into implementation, including software engineers, architects, data engineers, and solution designers working with real-world systems.

It is also relevant for technical leaders and decision-makers evaluating where digital twins can deliver value. If you are shaping strategy, selecting technologies, or guiding teams through complex system design, this book will help you understand the key opportunities and tradeoffs.

No prior experience with digital twins is required, but a basic familiarity with software development and distributed systems will help you get the most out of it. If you

are working on industrial or enterprise-scale twins, you will find that the core techniques covered here apply directly, while the text flags where additional considerations around governance, security, and organizational complexity come into play. Above all, this book is for builders who want a practical path to creating systems that reflect and interact with the real world.

How this book is organized: A road map

The book is organized into ten chapters that follow the lifecycle of a digital twin project.

It begins by introducing digital twin concepts, architectures, and industry applications before exploring how you can define measurable objectives and model physical systems. It then covers sensing and data collection, data integration and storage, and semantic modeling using ontologies and knowledge graphs.

The middle chapters focus on visualization in both 2D and 3D, followed by the use of AI/ML for prediction and automation, and simulation techniques for what-if analysis.

The book concludes with the practical realities of deploying and operating digital twins in production, including governance, security, monitoring, and long-term maintenance.

Although the chapters build on one another, each is designed to be largely self-contained, allowing readers to focus on the areas most relevant to their interests or current projects.

About the code

The book contains samples of source code both in numbered listings and in line with normal text. In both cases, the code is formatted in a fixed-width font like this to distinguish it from ordinary text. In many cases, the source code contains only the essential component to illustrate the concept, with full listings being available on GitHub. The code is predominantly in Python, with frontend code being HTML, CSS, and JavaScript. I have deliberately avoided the use of frontend frameworks to keep the examples concise.

The cloud services used in this book are from Amazon Web Services (AWS) because it is the platform I know best. The concepts and architectural patterns are not AWS-specific. Microsoft Azure, Google Cloud Platform, and other providers offer equivalent capabilities for IoT data ingestion, serverless compute, storage, and machine learning.

Some examples depend on data from IoT sensors, images, and other data. Where this is the case, sample data from my home digital twin is provided in GitHub so you can run the example code without needing your own IoT sensors. Many visualization concepts are best served with practical examples that are difficult to convey in a printed book. A website containing these examples is maintained with links provided in the text.

All the code samples in the book are available on GitHub at <https://github.com/digital-twins-in-action/book>. I use an ARM-based Mac, and all samples have been tested on that architecture (apart from the Modelica FMU that depends on Linux x86 architecture). The <https://www.digital-twin-book.com/> site will continue to be developed with resources that will be useful to you on your digital twin journey.

You can get executable snippets of code from the liveBook (online) version of this book at <https://livebook.manning.com/book/digital-twins-in-action>. The complete code for the examples in the book is also available for download from the Manning website at www.manning.com/books/digital-twins-in-action.

liveBook discussion forum

Purchase of *Digital Twins in Action* includes free access to liveBook, Manning's online reading platform. Using liveBook's exclusive discussion features, you can attach comments to the book globally or to specific sections or paragraphs. It's a snap to make notes for yourself, ask and answer technical questions, and receive help from the author and other users. To access the forum, go to <https://livebook.manning.com/book/digital-twins-in-action/discussion>.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the author can take place. It is not a commitment to any specific amount of participation on the part of the author, whose contribution to the forum remains voluntary (and unpaid). We suggest you try asking the author some challenging questions lest his interest stray! The forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

about the author



GREG BIEGEL has been working as a researcher, software developer, architect, and product manager across academia, startups, and large enterprises for the past 26 years. His fascination with bridging the physical and digital worlds began during his research into context-aware computing, where he first explored the early potential of IoT and software agents.

Greg is currently a Solution Architect with Amazon Web Services, based in Perth, Western Australia. He can be found on LinkedIn at www.linkedin.com/in/gregbiegel/.

about the cover illustration

The figure on the cover of *Digital Twins in Action* is “Musicienne du sérail” or “Musician of the harem,” taken from George Arents Collection, Album of Turkish Costume Paintings (1808–1826). Each illustration is finely drawn and colored by hand.

In those days, it was easy to identify where people lived and what their trade or station in life was just by their dress. Manning celebrates the inventiveness and initiative of the computer business with book covers based on the rich diversity of regional culture centuries ago, brought back to life by pictures from collections such as this one.

Bridging the physical and digital worlds

This chapter covers

- Defining what a digital twin is
- Different levels of digital twin maturity
- What digital twins are good for and when to avoid them
- How digital twins are used across industries
- Considerations when embarking on building a digital twin

Digital twins are virtual representations of physical systems connected to real-world data, enabling them to reflect current conditions and inform predictions about future behavior. The concept builds on decades of simulation and systems modeling, including work at NASA during the Apollo program, although the term itself emerged later, most notably through work in product life cycle management in the early 2000s. What distinguishes digital twins from the earlier versions is their persistent, operational data linkage to the physical systems they represent, which allows for continuous synchronization rather than one-time or periodic simulation.

For years, digital twins remained accessible only to large, well-funded organizations that could afford the scarce, expensive sensors, computing capacity, and advanced analytics required to build them. That's all changed. Today, digital twins can be built by almost anyone, thanks to the wide availability of low-cost sensors, high-performance pay-as-you-go computing, and powerful artificial intelligence and machine learning tools.

Some people think of digital twins as detailed 3D models, while others see them as glorified dashboards or traditional simulation systems. A digital twin may include elements from any of these, but it is not defined by any single one. At its core, a digital twin is a fit-for-purpose digital representation of a physical system that remains synchronized with real-world data and serves a defined analytical or operational objective. It does not need to be a high-fidelity 3D replica. A statistical model, a graph of relationships, or a set of structured equations can all qualify, provided the representation is meaningfully linked to the physical system and supports the decisions it was built to improve. More advanced implementations may add simulation, optimization, or bidirectional actuation, but these are capabilities that extend the twin, not prerequisites that define it.

What exactly counts as a digital twin? Let's start with a clear definition before we explore why they've become so popular.

1.1 *What is a digital twin?*

Some might say that the very last thing the world needs is another definition of a digital twin. But to build something meaningful, we have to be precise about what we mean. We aren't just building a 3D dashboard or an elaborate spreadsheet, but a living system that ingests data from its physical counterpart, maintains state, and represents the relationships between components of a physical system. For the purposes of this book, a digital twin is defined as follows.

Digital twin: A digital representation of a physical system that is updated with data from the real world. It helps users understand current conditions, monitor performance, and simulate scenarios to make better decisions based on clearly defined objectives. Digital twins may also act on the physical system by sending instructions to adjust or optimize its behavior.

This definition emphasizes that digital twins should not be built merely because the technology is available; they should serve specific business, operational, or research goals. By providing a virtual model that mirrors something physical, they help users comprehend both the state of the object or system and its structure. By remaining synchronized through continuous updates about current conditions, digital twins allow users to track the behavior of the physical system while also predicting the effect of any changes without changing the real system.

Figure 1.1 shows the three main elements of any digital twin: the physical system itself, its representation in software, and the flow of data from the physical system to the virtual one, with information, decisions, and actions flowing back.

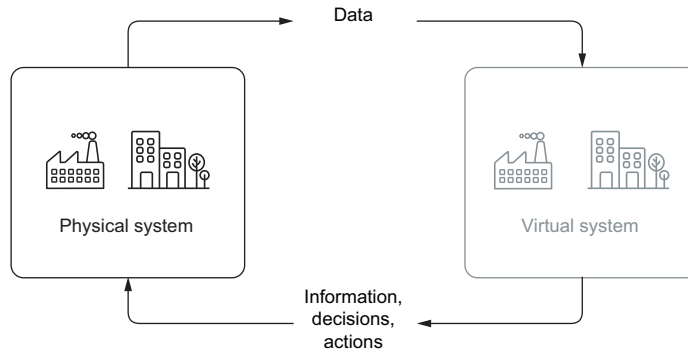


Figure 1.1 The three main elements of a digital twin are the physical system, the virtual system, and the flow of data and information between them.

NOTE Some definitions distinguish between a digital model (a static representation), a digital shadow (a model updated by a one-way data flow from physical to digital), and a digital twin (with a bidirectional data flow between physical and virtual systems). In practice, industry often uses digital twin as an umbrella term for all three. This book adopts that broader view while recognizing that each represents a different level of integration and feedback between physical and digital systems.

For most of human history, you had to be physically present to understand how something worked. Digital twins have changed that, and today, countless people can simultaneously monitor, analyze, and run simulations on the same physical system from around the globe.

1.2 Technologies enabling digital twins

Consider a wind farm. Before any turbines are installed, engineers use detailed simulation models, sometimes called virtual prototypes, to test turbine designs under a range of anticipated conditions, refining blade geometry, structural tolerances, and control strategies without the cost of physical prototyping. These models are powerful, but they are not yet digital twins in the full sense because there is no physical counterpart to synchronize with. Once the turbines are deployed, internet-of-things (IoT) sensors stream data to the cloud, where artificial intelligence (AI) or machine learning (ML) algorithms detect early signs of bearing wear. The digital twin then simulates maintenance scenarios and schedules repairs while simultaneously adjusting blade pitch across the farm to maximize energy capture based on real-time weather patterns.

This progression from virtual prototype to operational digital twin illustrates the full technology stack that underpins modern digital twins: IoT sensors capture physical data, cloud computing provides scalable processing and storage, and AI/ML generates predictions and drives optimization. It also highlights three distinct capabilities that emerge at different stages: virtual modeling during design to reduce physical

prototyping, predictive maintenance to prevent expensive failures once the system is operational, and near-real-time optimization to maximize ongoing efficiency.

Digital twins first gained traction in capital-intensive industries like manufacturing, utilities, and energy. These industries require massive investments, often operate in remote or hazardous environments, and face strict safety and regulatory requirements. Although these industries have used operational technology (OT) systems, like programmable logic controllers (PLCs) and supervisory control and data acquisition (SCADA), for monitoring and control since the 1960s, digital twins represent a major leap forward by integrating these traditional systems with modern IoT, cloud, and AI capabilities.

1.2.1 *Internet of things*

Not too long ago, measuring things such as temperature, vibration, location, or air quality required bulky, expensive equipment. Today's affordable, compact sensors enable users to capture detailed data about virtually any physical environment or system, opening up unprecedented possibilities for monitoring and control. Hobbyists can now build sophisticated home automation systems that measure everything from air quality to energy usage, while startups can deploy sensor networks to monitor equipment performance or environmental conditions at a fraction of the historical cost.

NOTE The DHT22 sensor I use to measure temperature and humidity in my home costs around \$5 today. Three decades ago, comparable digital measurement required industrial sensors and data acquisition systems costing thousands of dollars and were accessible only to well-funded organizations.

Large enterprises use these same technologies to monitor manufacturing lines, perform predictive maintenance, and optimize resources. Heavy industries have relied on SCADA and PLC infrastructure for decades, and in many such environments, those established systems still carry most of the real-time data flow, with newer IoT devices representing a growing but relatively small share.

In this book, we'll use the term IoT broadly to encompass the full spectrum of real-time sensing and data acquisition that feeds a digital twin, from low-cost wireless sensors to industrial control systems. The sensors already embedded in smartphones demonstrate this power by measuring motion, orientation, and light levels, and even performing 3D spatial mapping, turning everyday devices into powerful data collection tools. Figure 1.2 shows output from an iPhone's three-axis accelerometer while it's carried in a pocket; the distinctive pattern in this data allows us to infer that the person carrying it is walking. This democratization of sensing technology means anyone can now gather the rich, continuous data streams needed to build digital twins, which once required massive budgets and specialized expertise.

Gathering measurements is only half the story; the data must be sent somewhere useful. Advances in wireless communications have matched this sensor revolution. Fifth-generation (5G) cellular networks offer high speeds for high-bandwidth data. In contrast, low-power wide area networks (LPWANs) like LoRaWAN can transmit over

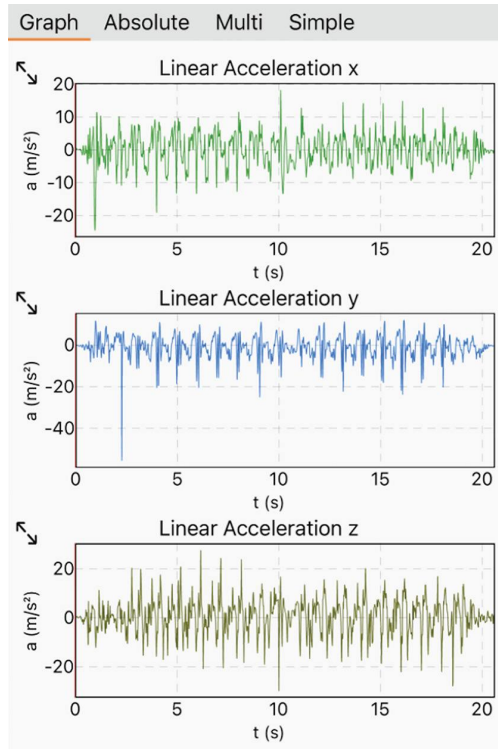


Figure 1.2 The phyphox app on an iPhone shows output from a three-axis accelerometer indicating that the owner is walking with the phone in their pocket. Such sensor data enables a digital twin to mirror and interpret real-world motion and behavior in real time.

10 km (compared with Wi-Fi's 50–100 meters) while using 10 to 20 times less power than Wi-Fi. This efficiency enables sensors to run for years on a single battery, making them ideal for small, resource-constrained devices deployed globally. Lightweight messaging protocols are used alongside these networks to transmit sensor readings efficiently from constrained devices to edge and cloud services.

IoT combines widely available commodity sensors and actuators (“things”) embedded in objects and connected to communications networks. As you move through the book, you’ll learn how to wire up a fleet of LoRaWAN-based sensors, route their data through a cloud pipeline, and use it to update a digital twin model in near real time.

1.2.2 Cloud computing

In the past, digital twins were only deployed by large industrial players because of the massive upfront investment required for hardware (data storage, analysis, simulation, and 3D rendering). Today, the public cloud removes that barrier, not because a digital twin must run in the cloud, but because the cloud provides access to compute, storage, and specialized services on a pay-as-you-go basis. This allows organizations of any size to scale resources to match their needs without significant capital expenditure: compute resources for complex simulations, elastic storage for continuous sensor data streams, and managed services for data processing, analytics, and machine learning.

This model removes the traditional financial constraints, allowing for rapid prototyping and scaling.

Edge computing complements cloud computing by processing data closer to where the data is generated, typically on local devices or gateways near the physical system. For industrial equipment or remote assets, edge devices perform local analytics, filtering, and decision-making in milliseconds, reducing latency for time-critical operations and minimizing bandwidth costs. For example, a security camera can run a machine learning model to detect defects or safety violations on a production line and transmit only the alerts and metadata instead of streaming hours of raw video footage to the cloud.

In this book, we'll look at how you can use some of the same cloud services that industrial organizations use today to help you build a digital twin.

1.2.3 *Machine learning and artificial intelligence*

The massive datasets generated by IoT sensors have limited value until they are filtered, stored, and analyzed to produce actionable insights. Recent advances in ML and AI, accelerated by specialized GPU hardware and the availability of large training datasets, have transformed these technologies from research tools into essential components of modern digital twins. Two categories of AI are particularly relevant here:

- Traditional machine learning involves algorithms that learn patterns from historical data to make predictions or decisions using classification, regression, anomaly detection, and time-series forecasting. When I refer to ML, this is generally what I mean.
- Generative AI is a newer class of models, including LLMs, and it creates interesting possibilities, including natural language interfaces for querying a twin's state, automated generation of simulation scenarios, and synthetic data creation when real-world data is scarce.

Both categories support the three capabilities that matter most in the digital twin context:

- *Pattern recognition and anomaly detection*—Identifying unusual behavior in continuous sensor streams that might indicate equipment degradation, process drift, or safety concerns before they escalate into failures
- *Predictive modeling*—Forecasting future system behavior based on historical patterns and current conditions, enabling proactive rather than reactive decision-making
- *Optimization*—Evaluating multiple possible configurations or actions in order to recommend or implement the best course of action based on defined objectives

Cloud providers have significantly lowered the barrier to entry by offering automated ML platforms that handle much of the complexity of building, training, and deploying models, from preprocessing sensor data to selecting algorithms and optimizing performance. For specific industries, specialized platforms offer prebuilt solutions tailored to

common use cases. When real-time responses are critical or cloud connectivity is unreliable, edge computing allows ML inference to happen locally on the device or gateway, enabling immediate decision-making without round-trip latency.

These capabilities are not confined to the cloud. We will explore how you can download pretrained models for common tasks such as object detection and time series forecasting, customize them with your own data, and run them locally. This gives you full control over your data and the ability to operate when internet connectivity is unavailable.

1.2.4 Visualization technologies and game engines

Sensors, IoT platforms, cloud systems, and AI models generate the data that powers a digital twin, but many users never interact with these technologies directly. Digital twins that include a visual component increasingly rely on game engines such as Unity and Unreal to render large-scale 3D environments in real time. Immersive technologies, including virtual, mixed, and extended reality, extend this further by allowing users to interact with data spatially, such as by walking through a virtual facility or overlaying live sensor data on physical equipment.

1.2.5 Agentic AI and autonomous decision-making

Agentic AI represents a shift toward systems composed of software agents that autonomously reason, plan, and execute complex workflows across digital twins. Unlike traditional automation, which follows predefined rules, agentic AI systems can decompose high-level objectives into subtasks, determine the best sequence of actions, and adapt their approach as conditions change. These agents can monitor systems, learn from data, make decisions, and perform actions to achieve goals, such as reducing costs or optimizing operations, either individually or in collaboration with other agents.

The relationship between digital twins and agentic AI is symbiotic. The digital twin provides the agent with a rich, contextualized representation of the physical system: its current state, historical behavior, and predicted future conditions. The agent, in turn, enhances the twin by continuously analyzing this data, identifying optimization opportunities, and executing actions that would be impractical for human operators to perform at the same speed or scale.

The promise of agentic AI comes with a corresponding need for safeguards. In industrial settings, autonomous actions can have safety, financial, and regulatory consequences that demand human oversight. Most production deployments operate on a spectrum of autonomy: an agent may act independently for low-risk adjustments (such as dimming lights in an unoccupied zone), recommend actions that require human approval for moderate-risk decisions (such as rescheduling a maintenance window), and be prohibited from acting without explicit authorization in safety-critical domains (such as adjusting pressure in a chemical reactor). Defining these boundaries and encoding them into the system are as important as building the agent itself. Readers working in safety-critical or regulated environments should consult domain-specific standards such as IEC 61508 for functional safety (www.iec.ch/functional-safety) and

the NIST AI Risk Management Framework (www.nist.gov/itl/ai-risk-management-framework).

1.3 *Digital twin capability levels*

When you hear the term “digital twin,” you might picture a sleek 3D visualization of a complex machine. Although visualizations are a common component, they are not the whole story. In fact, some digital twins operate as headless systems focused purely on data processing, analytics, and automated decision-making, without any visual component at all.

Digital twins range from simple monitoring systems to complex predictive and autonomous models. Classifying them by maturity will help you identify the right goal for your needs and gauge the expected business value. This book follows a five-level classification adapted from a model developed by Verdantix in a 2019 report titled “Smart Innovators: Digital Twins for Industrial Facilities.” It categorizes digital twins on a scale from basic descriptive systems to advanced autonomous operations, as shown in figure 1.3.

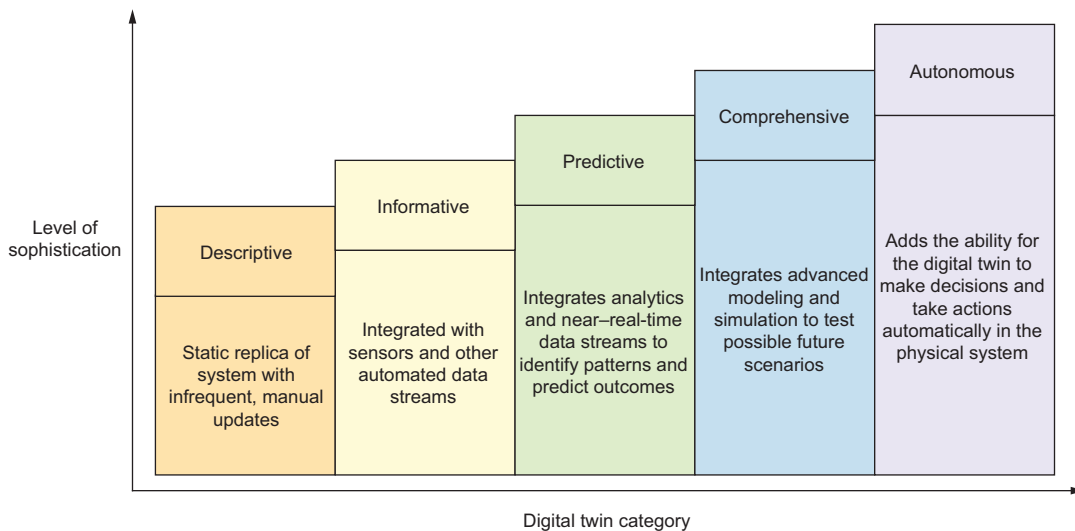


Figure 1.3 The five categories of digital twins

A few important caveats apply to any classification like this one. First, these levels describe capabilities, not a single linear path that every digital twin must follow. A system may be highly advanced in one dimension, such as simulation fidelity, predictive accuracy, or data integration, without necessarily being autonomous. Maturity, capability, complexity, and autonomy are related but not equivalent, and conflating them can lead to misguided design decisions.

Second, most implementations don't sit neatly at a single level. A system might combine real-time monitoring with failure prediction while still requiring human approval for actions. The classification is a lens for thinking about what a twin can do, not a rigid hierarchy where higher automatically means better.

NOTE Another useful approach to defining the capabilities of a digital twin is the Digital Twin Capabilities Periodic Table, published by the Digital Twin Consortium: <https://mng.bz/xZ8B>.

1.3.1 Descriptive digital twin

At the most basic level is the *descriptive digital twin*. This twin provides a reasonably static digital representation of a physical object or system, without making real-time predictions or responding to changes. This representation might be a simple engineering diagram or a fully rendered 3D visualization. In this type of digital twin, synchronization with the physical system occurs only infrequently. The model is updated periodically with new measurements or operational data, but it does not maintain a continuous connection to the real system and therefore reflects its state only at specific update intervals rather than in real time.

REAL-WORLD EXAMPLE: ONLINE MAPS

A familiar example of a descriptive digital twin is Google Maps. When you use this product, you see a digital model of real-world cities, containing roads, buildings, and parks, generated from cartographic maps, satellite imagery, and other sources, as shown in figure 1.4.

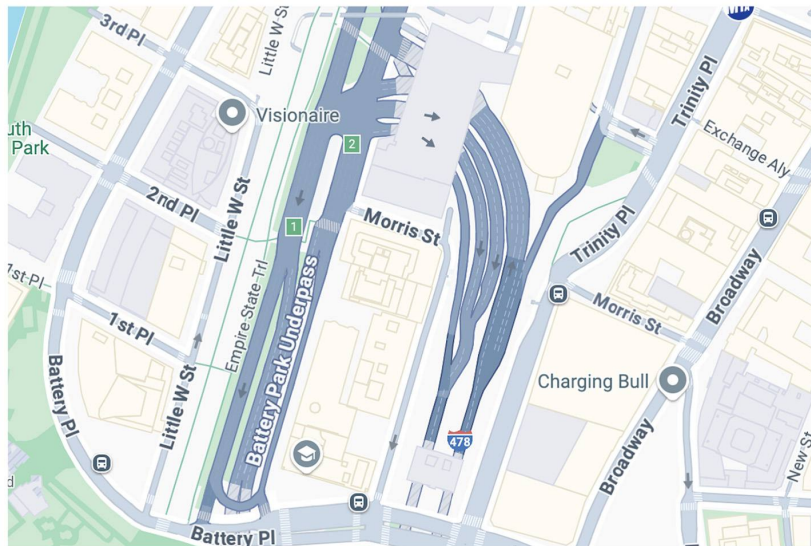


Figure 1.4 A Google Maps view of lower Manhattan, a familiar example of a descriptive digital twin. Map data © 2025 Google. Google Maps is a trademark of Google LLC.