Clausen *et al. Energy Informatics* 2022, **5**(Suppl 4):42
https://doi.org/10.1186/s42162-022-00222-7

Energy Informatics

RESEARCH

Open Access

Can we benefit from game engines to develop digital twins for planning the deployment of photovoltaics?

Christian Skafte Beck Clausen*, Zheng Grace Ma and Bo Nørregaard Jørgensen

From Energy Informatics.Academy Conference 2022 (EI.A 2022)
Vejle, Denmark. 24-25 August 2022

*Correspondence:
csbc@mmmi.sdu.dk

SDU Center for Energy Informatics, Maersk Mc-Kinney Moller Institute, University of Southern Denmark, 5230 Odense, Denmark

Abstract

Digital Twins (DTs) have attracted great attention in the energy sector. Game engines have been suggested to model DTs of their physical counterparts because they provide realistic graphics, lighting-, fluid- and physics engines that simulate the real world. However, the application of game engines to develop DTs for photovoltaics (PVs) has not yet been discussed in the literature. Therefore, this paper assesses the built-in game engine features' ability to support the DT development of PVs with Unreal Engine 5. This paper mainly focuses on visual representation because the surrounding environment significantly impacts PV deployment, and the existing software tools do not allow the study of the environmental factors at the early planning phase of a project's lifecycle. Furthermore, this paper investigates the position of the sun, shadows and reflections from nearby objects that influence the PVs' power output, and if the built-in light engine can be used for planning the deployment of PVs. The result shows that in-game objects in the environment can be used to affect the simulated PV output estimate over a year. It also indicates that applying Unreal Engine 5 to model PV systems that rely on mirroring real-world behaviour is promising if accurate data is used in the modelling. Real data and mathematical PV models are necessary since Unreal Engine 5's Lumen subsystem cannot provide realistic solar radiance on PVs for a given location on earth.

Keywords: Digital twin, Photovoltaics, Game engine, Unreal Engine, Simulation

Introduction

Photovoltaics (PV) are one of the most important renewable energy resources in the energy systems (Ma et al. 2019a). The scale of PV applications ranges from residential to utility deployments and has already been applied successfully in many regions worldwide e.g., (Ma et al. 2018; Billanes et al. 2018). The global capacity has increased from 100 Gigawatt (GW) in 2012 to 1 Terawatt (TW) in 2022, and it is estimated that this trend continues (Schmela 2022). During the same period, the price of solar modules decreased by approximately 90%. The decreasing investment costs and price pr. kWh, combined with efforts to reduce the carbon footprint make PVs favourable when investing in renewable energy sources (Ma et al. 2019b). However, PV deployment still faces multiple challenges, e.g., geometry such as landscape, layout, mounting and absolute position on earth.

Economic models, simulation- and optimisation software have addressed many of these challenges by using historical weather data, synthetic solar data, and solar radiance incident models that predict [W/m^2] on fixed and tracking surfaces, e.g., HOMER, Aurora, Bluesol and PV Sol. The proposed modelling, simulation and software are usually used in the early planning phases to determine the feasibility of the PVs in an area. Furthermore, visualization is typically included to provide the means of plots, 2D schematics, satellite views, or low poly 3D graphics.

In recent literature, mirroring physical systems in software constitutes the concept of a "digital twin" (DT). A DT is a representative virtual model that reflects a physical object or process. A DT differentiates itself from simulation by utilizing a real-time bi-directional connection between the virtual model and the physical object or process. This connection enables instantaneous updates from the real world to the virtual model that informs the underlying model and thus improves its capabilities. Furthermore, the virtual model can make corrective changes to the physical object or process by manipulating its controls. Making corrective changes aims to improve productivity and reduce operating costs and failure rates. DTs have attracted significant attention in the energy sector; much literature has discussed the application of DTs for PVs, such as digital twin modelling of photovoltaic panels (Delussu et al. 2022) and digital twins of solar farms (Arafet and Berlanga 2021).

With the recent advancements in state-of-art game engine frameworks, new opportunities arise. Game engines provide photorealistic graphics, lighting-, fluid- and physics engines that simulate the real world. Game engines offer a highly interconnected simulator for responding to real-time input locally or across computer networks. These features make them suitable for studying and exploring how next-generation PV systems can be developed by exploiting these powerful tools. Furthermore, game engines support incremental development where early models can gradually expand. This gradual expansion supports the project lifecycle from modelling early prototypes to eventually mirroring the physically deployed PV system. However, the application of game engines to develop digital twins for PVs has not yet been discussed in the literature. Whether it is possible to benefit from game engines to build DTs for planning the deployment of photovoltaics remains unclear.

Therefore, this paper aims to assess game engines as a tool to develop a DT for PVs. The overall objective is to assess to which degree the built-in game engine features support the DT development of PVs. This paper mainly focuses on visual representation because the existing software tools do not allow to study the environment with a high degree of presentability.

Furthermore, this paper investigates if the built-in components in UE5 can be used for planning the deployment of PVs and its power output. The basic premise is the capability to represent the position of the sun in the sky visually correct in relation to the position on earth and the time of year. PVs depend on solar irradiance [W/m^2], which requires that the game engine must be able to provide this value. To assess the features of the small-scale residential PV system, the configuration of the PVs, the solar irradiance and the power output estimate of the PVs are implemented.

The rest of the paper is structured as follows: LITERATURE REVIEW presents the state-of-art and current limits within digital twins and their applications. METHODOLOGY introduces the main steps of the process, the tools used to implement the prototype and related methods. PV MODELLING AND DIGITAL TWIN ARCHITECTURE DESIGN presents the design of the PV prototype. SCENARIO DESIGN presents each scenario under study. RESULTS presents the results of each scenario under consideration. DISCUSSION contains a discussion of the results and the advantages and limitations of applying game engines for planning digital twins of PVs. CONCLUSION concludes the paper.

Literature review

This section presents the origin and definitions of DT as stated by the literature. The definitions are used to establish the nomenclature used throughout this paper. Furthermore, the purposes, methodologies and application domains of DT are presented followed by a discussion of the challenges.

The definitions of digital twins

The concept of DTs was anticipated in 2002 by Grieves in a presentation at the University of Michigan, where it was introduced as a conceptual idea for the product lifecycle management (Grieves et al. 2017). DTs have received increased scientific attention since NASA's Technology Roadmap Report in 2010 (Piascik et al. 2010). NASA elaborates with a DT being a paradigm that integrates technologies into a multi-physics, multi-scale simulation. An important aspect is that the DT utilize high-fidelity modelling and situational awareness to provide a real-time virtual construct. According to Grieves et al. (2017), a DT is defined as "... a set of virtual information constructs that fully describes potential or actual physical manufactured product from the micro atomic level to the macro geometrical level. At its optimum, any information that could be obtained from inspecting a physically manufactured product can be obtained from the Digital Twin".

Kritzinger et al. classify DT into subcategories through their level of integration between the physical and digital counterpart (Kritzinger et al. 2018). The subcategory Digital Model (DM) is a digital representation with no link to the physical object. Consequently, no data flows from the physical object to the virtual object, and the use cases are attributed to, e.g., simulation, mathematical models or any other physical object models. The subcategory Digital Shadow (DS) extends the definition of the (DM) by an automated one-way data flow from the physical object to the virtual object. The state of the physical object is reflected in the virtual object. The subcategory DT extends the DS by replacing the one-way link with a bidirectional link that enables the virtual object to control the physical object.

A DT is a dynamic representation throughout the product's life cycle i.e., creation, production, operation, and disposal (Grieves et al. 2017). Grieves et al. distinguish the types of DT into Digital Twin Prototype (DTP) and Digital Twin Instance (DTI) (Grieves et al. 2017). The DTP describes a prototypical physical artefact containing the information necessary to define and produce the physical object. The information can include requirements, a fully annotated 3D model, bill of materials, bill of processes, bill of services and bill of disposal.

The DTI describes the actual physical object that the DT remains linked to for its entire lifetime. The DTI contains additional information, e.g., current and past components, maintenance information, and operational states captured from historical or actual sensor data or predictions. The DTP or DTI is operated within a Digital Twin Environment (DTE), an integrated multi-domain physics application space. In the creation phase, the DTE is used to predict the behaviour and performance of a physical object by utilizing the DTP. In the operation phase, the DTE is used to predict behaviour and performance by utilizing historical- and actual measurements provided by the DTI.

Purposes

DT may serve multiple use cases; however, the most common use cases are for visualization, monitoring, prediction, optimisation and control. The literature generally does not distinguish between visualization and monitoring, even though these acts are carried out at different stages in the product lifecycle. Visualization can be referred to as DTP in the creation phase, while monitoring can be referred to DTI in the operating and maintenance phase. Monitoring and visualization can be realized with instrumentation such as dashboards (Lee et al. 2021; Felemban et al. 2021; Granelli et al. 2021; Lin and Low 2020), custom user interfaces (Xu et al. 2021; Zhang et al. 2021; Viola and Chen 2020; Leinonen et al. 2019) or game engines (Lee et al. 2021; Abdallah et al. 2020; Sharotry et al. 2020; Negrin et al. 2021; Pereira and Ellman 2020; Eyre et al. 2018; Kong et al. 2020; Zhifeng et al. 2021; Leskovsky et al. 2020; Li et al. 2021a; Fan et al. 2021), and requires that the thing being monitored provides a data stream, e.g., from sensors. The data stream may be real-time, near real-time or historical. Real-time data presents the physical object as close to reality as possible, while historical data is used to put the real-time data in the context of previous events. Visualization technology such as CAD (Computer-Aided Design) tools, simulation tools and 3D modelling tools are commonly used in the creation phases while dashboards, custom 2D User Interface (UI), game engines, virtual reality and augmented reality are used in the operation phase.

Methodologies

The methods used to develop DTs span widely and depend on their purpose. Simulation methods are fundamental to constructing a DT to predict, optimise, and make decisions or detect faults. Simulation methods may be based on agent-based simulation (Belfadel et al. 2021; Meta et al. 2021; Park et al. 2020; Rodríguez-Aguilar and Marmolejo-Saucedo 2020; Vrabič et al. 2021) or discrete event simulation (Eyre et al. 2018; Rodríguez-Aguilar and Marmolejo-Saucedo 2020; Vrabič et al. 2021; Karakra et al. 2020; Negri et al. 2019, 2021; Marmolejo-Saucedo 2021; Hyeong-su et al. 2019). Agent-based simulation is used to model autonomous agents that fulfil their own decisions to satisfy a common goal, e.g., people in urban simulation (Belfadel et al. 2021; Meta et al. 2021; Christensen et al. 2019), supply-chain agents (Park et al. 2020; Howard et al. 2021), and highly specialized units (Værbak et al. 2019) and learning agents (Vrabič et al. 2021). Discrete event simulation is a simulation method that simulates events with a chronological event queue, e.g., production processes (Christensen et al. 2020a, 2020b), arriving patients (Karakra et al. 2020), drilling station (Negri et al. 2019, 2021), periodic decisions in supply chains (Marmolejo-Saucedo 2021) and robot events (Eyre et al. 2018).

The simulation models may be interconnected to provide its result to external programs to establish feedback loops (Eyre et al. 2018). Interconnection of simulation models is also referred to as (1) synchronized simulation (Negri et al. 2019, 2021) or (2) co-simulation where heterogeneous simulation models are interconnected through functional mockup interfaces (Perabo et al. 2020). Simulation tools applied for DTs are such as MATLAB, AnyLogic, OpenModelica, Open Simulation Platform and EnergyPlus (Howard et al. 2021).

Artificial Intelligence (AI) is commonly used to construct DTs and is used for predicting, detecting, making decisions, planning and diagnosing. The most common AI methods include machine learning (Vrabič et al. 2021; D'Amico et al. 2019; La Russa and Santagati 2020; Darvishi et al. 2021; Demirel et al. 2021; Greis et al. 2021; Zohdi 2020; Hafez 2020; Min et al. 2019), neural networks (Vrabič et al. 2021; Mourtzis et al. 2021; Wu and Li 2021) and genetic algorithms (Viola and Chen 2020; Negri et al. 2019, 2021; Zohdi 2021a). Software engineering methods are mainly applied to specify functional requirements and document the DT's structure and behaviour.

Application domains

DT has been applied in a wide range of domains. The most prominent domains include manufacturing, buildings, aviation, logistics, mechanical engineering, healthcare, and energy systems (Howard et al. 2021). In energy systems, DT for renewable energy systems has been applied within agrophotovoltaics (Zohdi 2021b) and energy analytics platforms (Abdallah et al. 2020; Zhang et al. 2021). For instance, (Zohdi 2021b) presents a DT framework and method that aims to provide a computational framework to design and deploy complex agrophotovoltaic systems rapidly. The method allows for tracking and optimising solar power flow in solar farm facilities and is based on a genomic-based machine-learning algorithm that configures the system with various parameters to arrive at an optimal system.

Abdallah et al. (Abdallah et al. 2020) assessed the potential of game development platforms for energy systems—including renewable energy—by implementing a DT of an energy analytics lab. The results showed that Unity 3D hold the potential for visualizing the lab through Virtual Reality (VR) and enables bidirectional communication for monitoring and controlling purposes. Zhang et al. (Zhang et al. 2021) proposed a DT hybrid simulation platform intended for use in engineering education for renewable energy. The DT platform consisted of an experimental subsystem, real-time digital simulation subsystem, graphical display, and connectivity to remote data. The results showed that the DT platform could reproduce the behaviour of most industrial systems in renewable engineering with good accuracy. Furthermore, the DT platform significantly reduced the costs of acquiring expensive hardware while offering flexibility.

Razo et al. (Razo et al. 2020) proposed a method to parametrize and simulate PV systems with limited or no a priori knowledge. Using on-time measured power, ambient temperature, and satellite-derived irradiance with a genetic algorithm, they created a precise DT of a PV system. Liu et al. (Liu et al. 2021) developed a DT for a photovoltaic power station where they utilized Unity 3D for real-time visual rendering and monitoring. Their geometry model was built in 3ds Max and imported into Unity 3D to form the PV arrays and the complete system design. Their DT supports bidirectional communication for real-time data feed from the PV to the DT and conversely from the DT to the PV for instructions to guide operation.

Challenges

Multiple challenges of DTs are stated in the literature. Developing DTs is a non-trivial affair that requires cross-functional and highly skilled labour (Eyre et al. 2018). The lack of a universal terminology and classification makes DTs difficult to comprehend and communicate the concept (Khan et al. 2020; Volkov et al. 2021; Autiosalo et al. 2020; Barth et al. 2020; Raes et al. 2021; Minerva and Crespi 2021; Zhang and Sun 2021). This issue propagates challenges of scoping (Shao and Helu 2020), ontology (D'Amico et al. 2019), and increased development costs (Felemban et al. 2021; Landahl et al. 2018; Shah et al. 2021). Difficulties in scoping are emphasized by the multi-modalities of DT, i.e., visualization, monitoring, prediction, and control. Another issue is that the term DT has been hyped and used diligently to envelope established methodologies and technologies, making it a challenge to classify software as a DT (Minerva and Crespi 2021).

A DT requires connectivity to the physical object(s), which involves equipment with IoT (Internet-of-Thing) sensors and actuators already in operation. Given the heterogeneous nature of IoT and big data means that integration, data sharing and management issues arise (Negrin et al. 2021; Fan et al. 2021; Negri et al. 2021; Perabo et al. 2021; D'Amico et al. 2019; Adamenko et al. 2020; Lehner et al. 2021a, 2021b; Li et al. 2021b). Middleware that eases integration exists but at the cost of architectural- and infrastructural complexity, increasing development costs.

Practical approaches to developing DTs in commerce and industry are still sparse. This calls for practically proven standards, frameworks, and reference architectures in targeted domains. Furthermore, proven technology and methodologies for developing and operating DTs remain challenging (Negrin et al. 2021; Fan et al. 2021; Karakra et al. 2020; Min et al. 2019; Barth et al. 2020; Raes et al. 2021; Adamenko et al. 2020; Michael 2021; Piroumian 2021).

In relation to the planning of PVs, the current tools lack realistic visual representations that include the surrounding environment, i.e., the sun, buildings, roads, weather, shadows, and reflections. This can be used to present various scenarios of planning PV deployment to non-technical stakeholders. The work in this paper will contribute by providing a visual representation that reflects the minimum necessities for presenting and developing DT in PV applications.

Methodology

This paper aims to investigate the potential of game engines to develop DTs of PVs. A game engine is essentially a real-time simulator. The game engine processes a continuous stream of input and manipulates its internal model and visual components accordingly. This manipulation causes a state change which is reflected as a visual response in the form of animated graphics. Modern game engines provide features that are compelling for other applications than games. The advancement of photorealistic graphics, light-, fluid- and physics engines make them useful for engineering use cases. Choosing a game engine depends on the requirements since each game engine has its advantages and disadvantages. A set of requirements for a minimum viable prototype of residential rooftop PV was identified, as shown in Table 1.

A DTP is developed since the proposed PV DT is a prototype of a physical artefact and takes place in the creation phase. The DTP is not connected to the real world and therefore categorizes as a DM with a simulated behaviour. This is appropriate since the prototype is intended as an early implication assessment of applying game engines in the PV domain.

LG375Q1C-V5 were selected to represent the PV modules. This module consists of 60 cells in a 6×10 configuration and produces 375 W under standard test conditions 1000 W/m^2 and 25 °C. Arranged in a 16-module PV array gives a cumulated capacity 6 kWp which is a typical dimensioned array for average households. In a real-world scenario, the PV power production depends on factors such as varying load, irradiance, and temperature. This paper assumes standard test conditions to narrow the scope on the potential of game engines. For the same reason, inverters and batteries are not considered.

So far, game engines have not been suggested to develop PV DTs. Unity and Unreal Engine 5 (UE5) frequently represent DTs virtually. Both game engines provide rich features such as development environments, connectivity, asset stores, plugins, landscape editors, importability of 3D models and programmability through either visual coding, C# or C++. UE5 provides all features free of charge until royalties of 5% apply at USD 1 M gross revenue, whereas Unity is free of charge with an annual revenue cap of USD 100 K. For visual appearance, UE5 is superior in environment lighting, including dynamic shadowing, reflections, and directional light sources (i.e., the sun). The visual appeal and representation of the DT PV application are essential; therefore, UE5 is chosen in this paper. To develop the PV model and DTP, besides UE5, the following tools or plugins are applied in this paper:

- UE5's Marketplace assets represent the residential house and environment.
- UE5's built-in static mesh editor for modelling custom PV modules.
- UE5's blueprint feature for adding behaviour to in-game actors and levels.
- UE5's level feature for representing individual scenarios.

Table 1. Prototype requirements

ID	Brief requirement description
Requirement 1	Represent a light source that visually imitates the sun and its position on the sky depending on the absolute position (latitude, longitude) on earth and the time of the year
Requirement 2	Represent an array of commercially available PV modules with each module having the following attributes: Multiple cells Fixed slope and orientation (azimuth) Flat mounted directly on roof
Requirement 3	A PV module cell must be capable of: Power production when exposed to direct solar radiation (beam radiation) Power production when exposed to indirect solar radiation and take account for shadows (diffuse radiation)

- UE5's plugin 'Sun Positioning Calculator' available from their Marketplace.
- Python and matplotlib for post-processing the collected simulation data.

PV modelling and digital twin architecture design

This section describes the model design and the architecture of the DTP. The physical model is depicted through domain modelling, and each concept is explained. Subsequently, the implementation details of the domain model are explained.

PV model design

The model in Fig. 1 depicts the basic geometry for a residential PV array mounted on a rooftop. The PV array has a slope, the angle between the horizontal plane and the module surface; $0° \leq \beta \leq 180°$. An angle greater than 90° means that the module faces the ground. Furthermore, the PV array has an azimuth which is the orientation of the module surface where due south is 0°, east negative and west positive; $-180° \leq \gamma \leq 180°$. The zenith angle is the angle between the vertical line and the line from the sun. A module is built up of individual cells, each capable of converting solar irradiance into power.

Figure 2 shows the domain model of the depicted rooftop PV. The sun emits energy (irradiance) and is collected by the PV array. The PV array is arranged through a series of modules, and each module is mounted on the roof. The deployment parameters of a module are slope and azimuth which affect the annual power production. The power output of each module depends on the absolute position on earth (latitude/

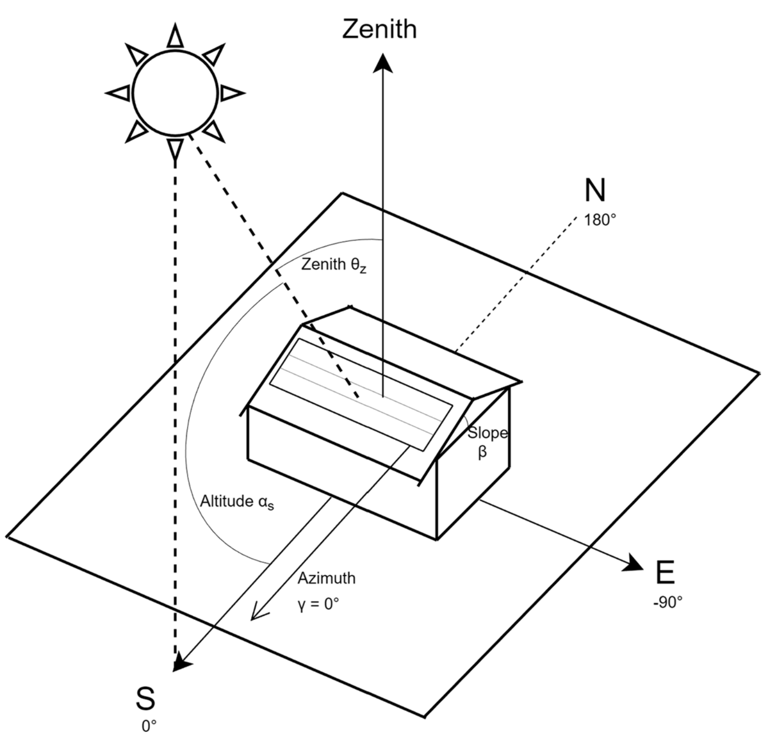

Fig. 1 Geometry of the PV domain

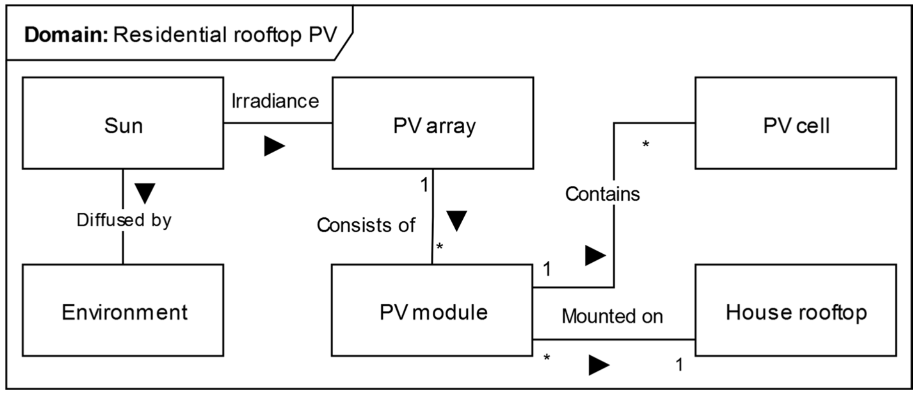

Fig. 2 PV domain model

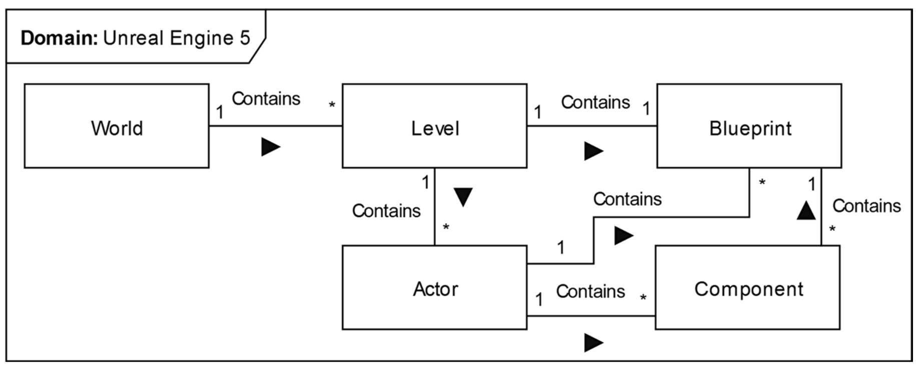

Fig. 3 UE5 meta-model

longitude), time of year, beam radiation, diffuse radiation, cell temperature, derating factor and ground reflectance (albedo).

The domain model is implemented in UE5. UE5 provides a framework of built-in types; therefore, the PV domain must be mapped into this framework (see Fig. 3). The UE5 meta-model is simplified to discuss the entities of relevance within this paper. UE5 is built up around the concept of a world in which multiple levels can exist. Actors are visual or non-visual objects that exist or can be spawned in a level, and they can be attributed with, e.g., static mesh, variables and behaviour. A static mesh is a non-animated geometry type that contains polygons which the game engine can render. Variables can be either primitive or complex types. Behaviour is realized through Blueprints, which are logic that can be attached to levels, actors, or components. A component is a generalized type that unlike an actor cannot be directly spawned in the world. A component must be attached to an actor, but the component is suitable for reuse across many actors.

Table 2 contains the mapping of entities from the PV domain to the UE5 meta-model. The SunSky actor enables the correct positioning of a sunlight source in the sky both at compile- and runtime. This functionality is provided out-of-the-box, and only the runtime behaviour must be implemented, i.e., positioning the sun by dynamically responding to event inputs.

Table 2 PV to UE5 domain mapping

PV domain entity	UE5 entity
Sun	The plugin 'Sun Position Calculator' (Geographically Accurate Sun Positioning Tool 2022) provides a SunSky actor which positions the in-game sun visually correct based on latitude/longitude, time zone and date/time
PV cell	Custom static mesh component
PV module	Custom actor with custom PV cell components and a glass material overlay
PV array	A series of custom PV module actors
House rooftop	Static Mesh actor that represents a residential house from the 'Modular Neighborhood Pack' downloaded from the UE5 Marketplace
Environment	Various static mesh actors to represent trees, roads, fences, bushes etc. downloaded from the UE5 Marketplace

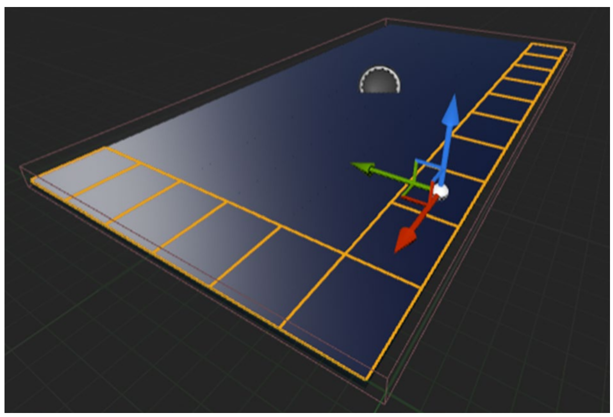

Fig. 4 Custom PV module with 60 cells arranged in 6×10 rows in UE5

Ideally, the PV modules should be a replica of the chosen commercial PVs. Unfortunately, such replica is unavailable. UE5 Marketplace provides realistically looking PV modules of fixed sizes. The prefabricated PV modules cost a small fee and does not come with 60 individual cells as the chosen modules. Therefore, modifications would have to be made. Creating a custom module involves a similar task of adding individual cells and behaviour through blueprints, but without additional fees. However, the look and feel of the PV module may not be as photorealistic. Deciding on the best approach is a trade-off on case-to-case basis. A custom PV module actor with 60 cells was implemented by representing each cell as a static mesh component of a cube and applying a blue glass material for visual effect (see Fig. 4). The PV modules can be arranged to form a PV array on a rooftop as seen in Fig. 5.

Detecting PV power production

UE5's Lumen Global Illumination system provides the option to use physical units (lux) to represent a scene realistically by reflecting and diffusing lights. Ideally, UE5 could be ingested with real-world measurements of solar irradiance of a specific location and then rely on the engine's direct- and reflected light and eventually be collected by the PVs. UE5 provides a dynamic histogram range (HDR) tool in editing mode for quantifying

Fig. 5 Rooftop with PV array consisting of 16 modules

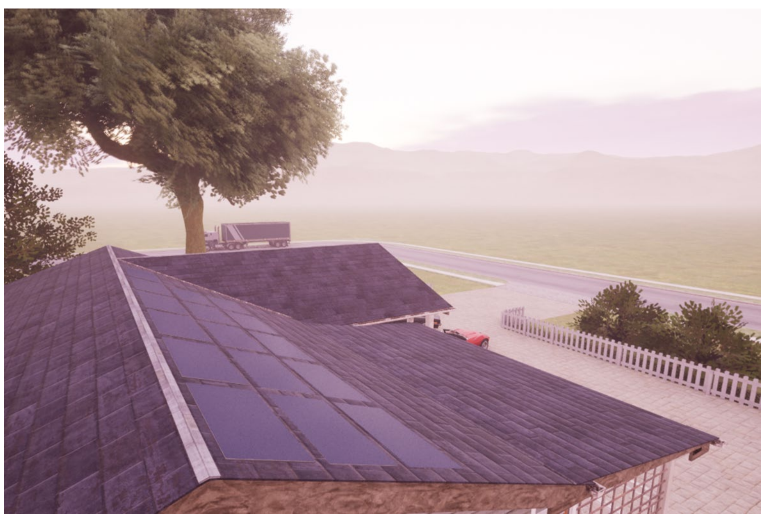

Fig. 6 PV array blocked by mountains in the horizon early in the morning

the reflected light from the camera's perspective. PV applications are interested in calculating radiation incidence on the PV array. However, in UE5, it is not possible to quantify the light that hits a given surface from the perspective of that surface. Instead, line tracing from the PV cells to the sun was used to determine if a direct beam hit the cells. In this way, it is possible to establish vectors from each PV cell to the sun to determine if these are within the line of sight. Any actor with enabled collision interrupts the line tracing. Figure 6 shows the sun being blocked by mountains on the horizon. The visible light on the PVs is therefore diffused. Figure 7 shows the line tracing from the individual PV cells to the sun when there is free line of sight.

Not all static mesh actors have collision by default. This is the case with trees where leaves are modelled with foliage. For these actors, the built-in static mesh editor was

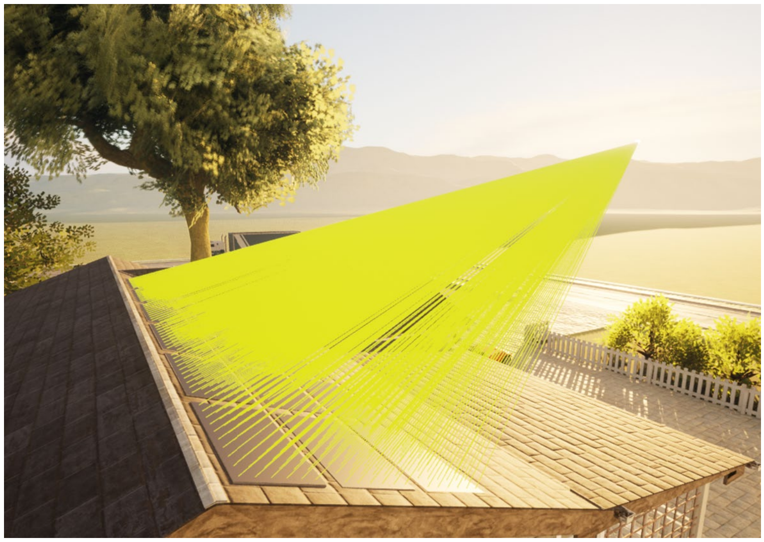

Fig. 7 PV array exposed to direct beam irradiation

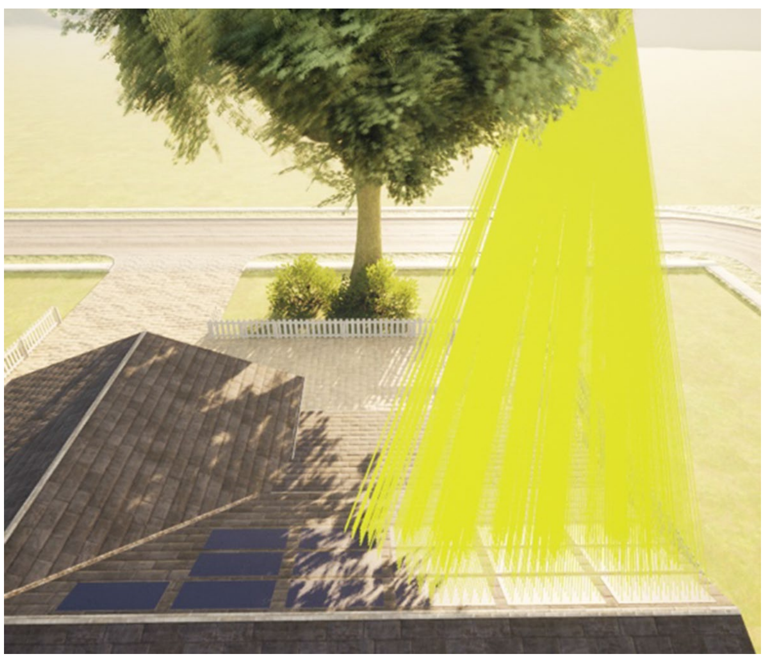

Fig. 8 A tree blocks line tracing

used to apply box- and capsule collision that formed the tree's crown. The effect is shown in Fig. 8 where there is no line trace on the shadowed panels. Figure 9 shows a sequence diagram of the cell activation mechanism. UE5 fires event ticks for each generated frame. The PVModule blueprint reacts by creating a line trace between the sun and each cell. The cell activates and provides full power output if there is a line of sight. Producing full power output on beam radiation does not reflect reality, but it is sufficient to assess the prototype. Encapsulating the activation mechanism in the PVModule makes it easy to scale the capacity by adding/removing PV modules to the rooftop.

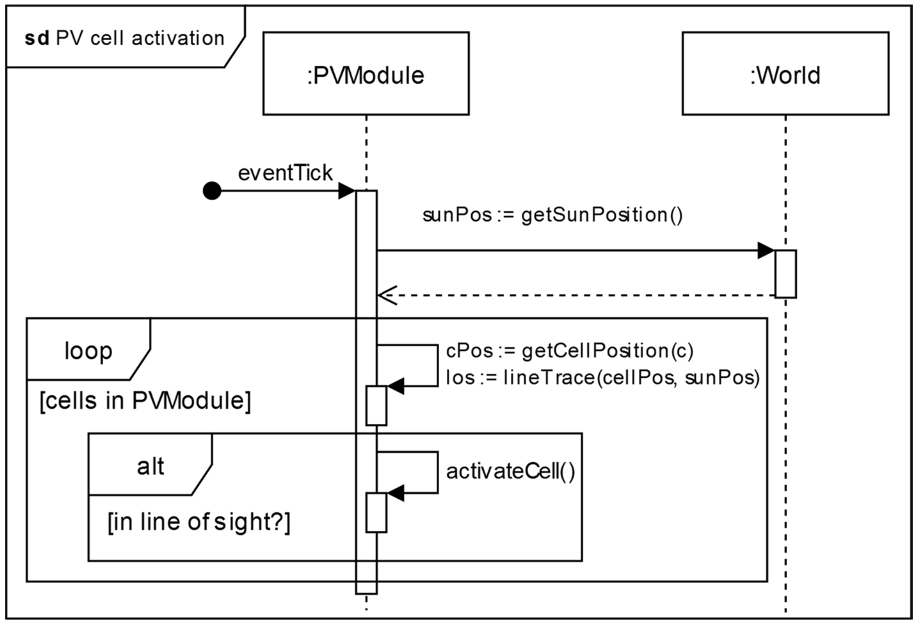

Fig. 9 Cell activation sequence diagram

Simulation model

A simulation model was implemented as a blueprint to simulate the PV power output for a year. The blueprint is illustrated in Fig. 10 is a simplified sequence diagram. The simulator is manually activated by a button in-game that initiates a Timer. A Timer is a built-in UE5 function that creates a separate thread and periodically executes a user-specified function. In this case, the Timer calls the method *simulate(from, to, resolution)*. Potentially, the call to simulate is a time-consuming process which blocks the frame generation if called for a whole year. Therefore, the period of the Timer defines the simulation speed.

The simulation speed must be set to a reasonable fraction of a second to allow UE5 to process frames. Fundamentally, the simulator samples the power output from the activated PV cells, adds the result to a time series and advances itself forward in time with the given resolution. Finally, the World is updated with the current time the Sun-Sky actor uses to position the sun. The time series is exported to a log file that can be retrieved for post-processing.

Scenario design

The implemented DTP of the rooftop PV is simulated in two scenarios with the simulation model:

The baseline scenario

- Purpose: The baseline scenario illustrates an ideal clear-sky environment where no objects block the PV array.

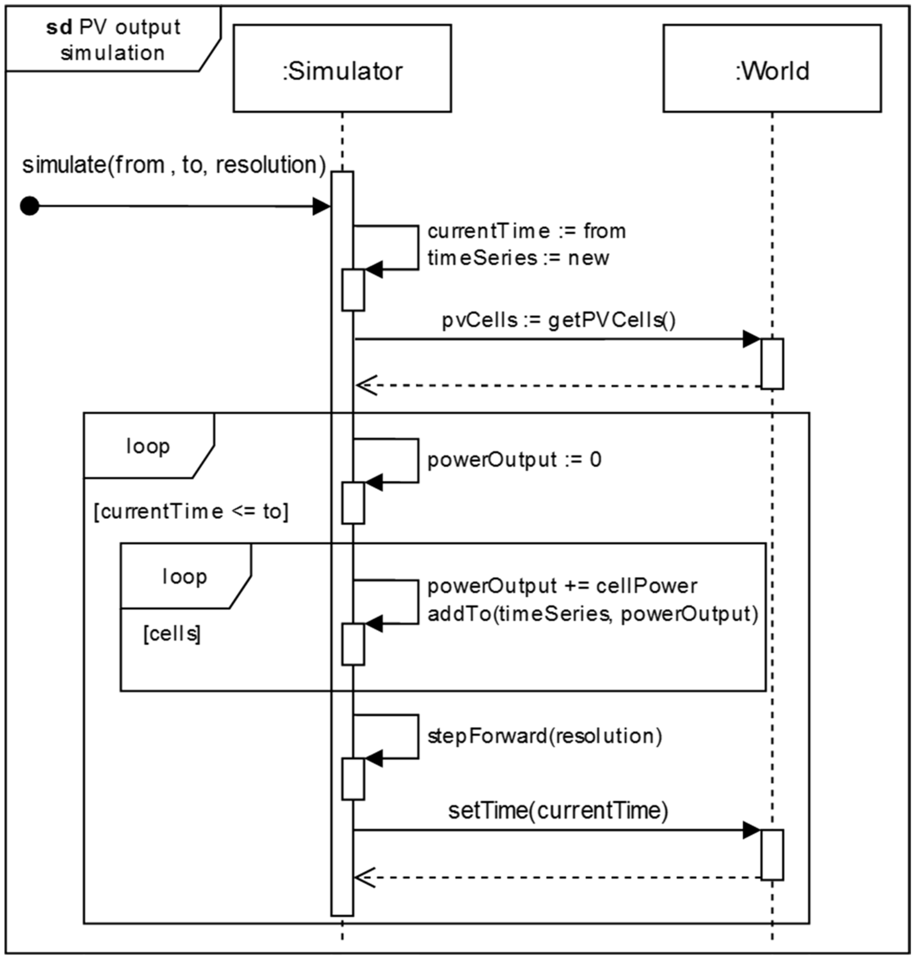

Fig. 10 Simulation model

- Description: The baseline scenario is a flat surface environment as shown in Fig. 11. The PVs are mounted on a rooftop with an azimuth of $\gamma = 0°$ and a slope of $\beta = 20°$, which corresponds to the slope of the roof. A single PV module represents a typical commercially available module with a capacity of 0.375 kWp.
- Data input: as shown in Table 3.
- Data output: The variable of interest is the PV array production pr. day [kWh/day] for a one-year duration. Each cell produces full capacity [W] when exposed to a direct beam, i.e., when the cell is activated. This output is represented as a floating-point number attributed to the given cell in any generated frame.

The alternative scenario

- Purpose: The alternative scenario illustrates the implication of objects that may block the direct beam radiation over the year.
- Description: The alternative scenario corresponds to the baseline clear-sky environment but with surrounding trees with collision boxes, as seen in Figs. 12 and 13. The

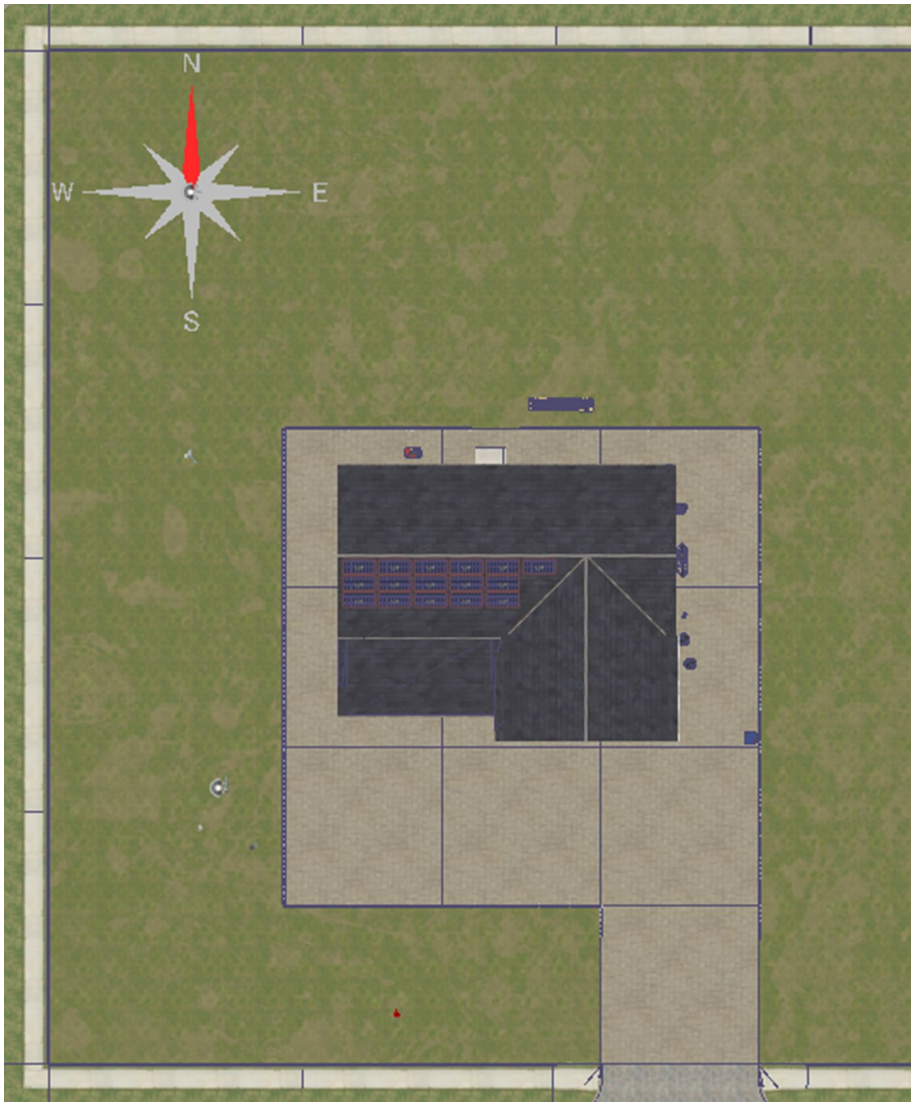

Fig. 11 The baseline scenario

Table 3 Data input baseline scenario

Parameter	Value
Azimuth	0°
Slope	20°
PV cell capacity	6.25 W
PV module capacity	6.25 W * 60 cells = 375 W
PV array capacity	375 W * 16 modules = 6000 W
Simulation start	2022-01-01 00:00:00
Simulation end	2023-01-01 00:00:00
Simulation step size	1 min
Simulation speed	10 ms i.e., 10 ms advances the simulation with the given step size in-game
Absolute position on earth	latitude, longitude
Time zone	UTC corresponding to the given latitude and longitude

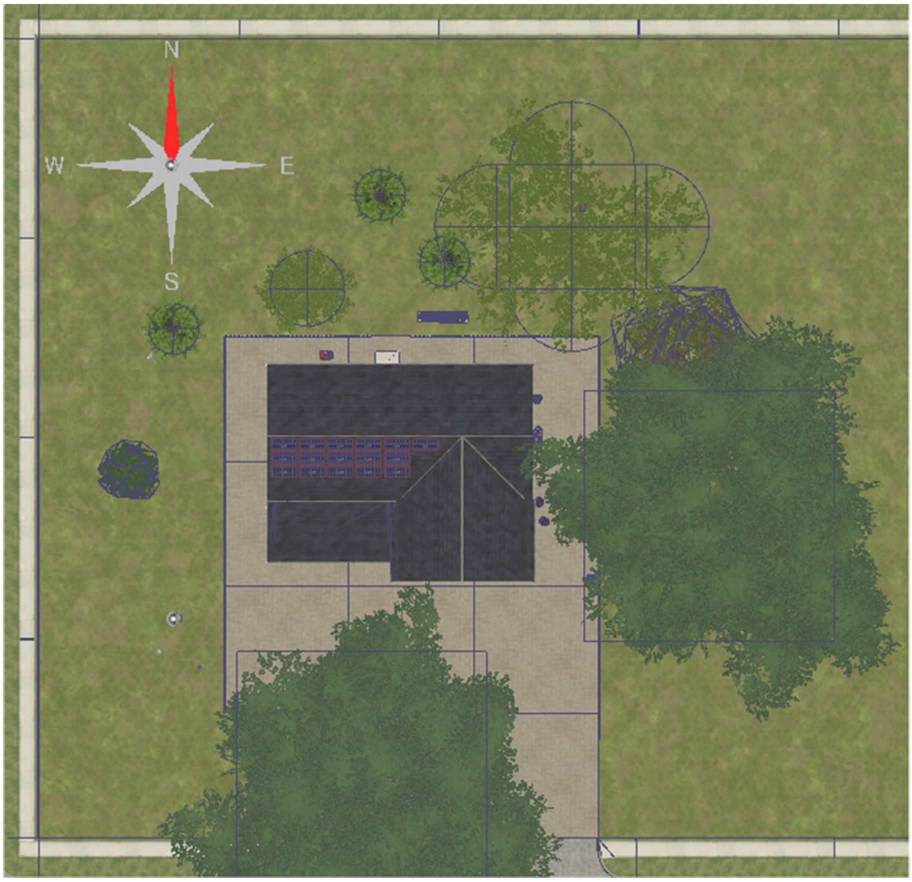

Fig. 12 The alternative scenario, top view

Fig. 13 The alternative scenario, side view

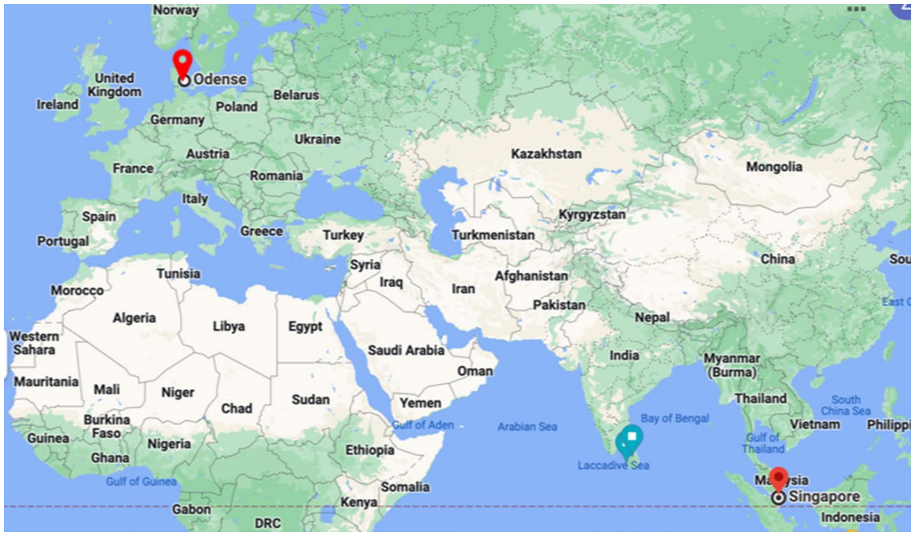

Fig. 14 Two chosen locations (Odense Denmark and Singapore)

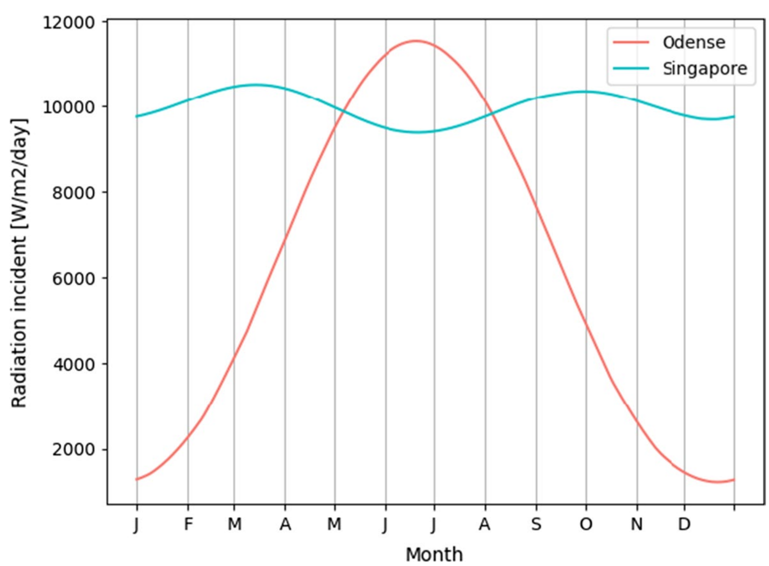

Fig. 15 Daily extraterrestrial radiation

exact dimensions and positions of the trees are unspecified since this information is not necessary to compare differences between the scenarios. Measurements can be retrieved through UE5's measurement tool. The choice of trees is for illustration purposes. Other obstacles could have been used, such as buildings and elevated landscapes if collision boxes are applied.

- Data input: Equal to the baseline scenario.
- Data output: Equal to the baseline scenario.

Two locations (latitude, longitude, shown in Fig. 14) on earth were chosen for testing the effect of the absolute position on earth in relation to the received solar radiation over

Table 4 Sample data from simulation

Timestamp	Watts
2022-01-23 09:27:00	5412.5
2022-01-23 09:28:00	5412.5
2022-01-23 09:29:00	5525
2022-01-23 09:30:00	5525
…	…

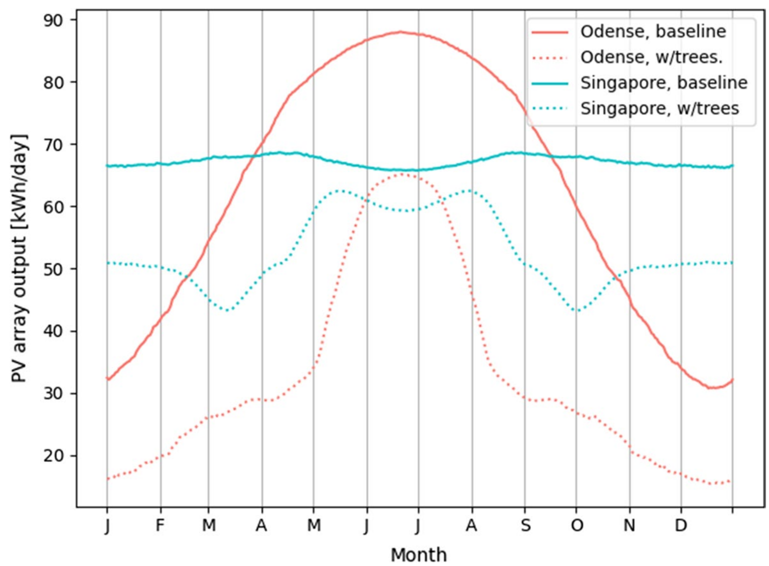

Fig. 16 The simulated daily PV array power output for all scenarios

a year. It is expected that the northern hemisphere receives concentrated radiation during summer and that radiation fluctuates less over the year near the equator (see Fig. 15). Singapore is located very near equator and therefore this location was chosen. Any location with a latitude close to equator would be sufficient.

- (55.403756, 10.40237), UTC + 1 Odense, Denmark.
- (1.290270, 103.851959), UTC + 8, Singapore.

Results

The results were conducted on a workstation machine consisting of an AMD Ryzen 9 5900X 12-core processor, 64 GB DDR4-3200 RAM, NVIDIA GeForce RTX 3060 and a Samsung 970 EVO NVMe drive. Each simulation ran for approximately 1.5 h. A sample of the collected data is shown in Table 4 with a timestamp and the watts produced in that frame. It is assumed that the number of watts produced within that minute remains unchanged; therefore, this value is equal to watt minutes [Wm].

The watt minutes were aggregated into kilowatt hours per day [kWh/day] by summarizing the 1440 min of each day and dividing by 60.000. Figure 16 shows the results

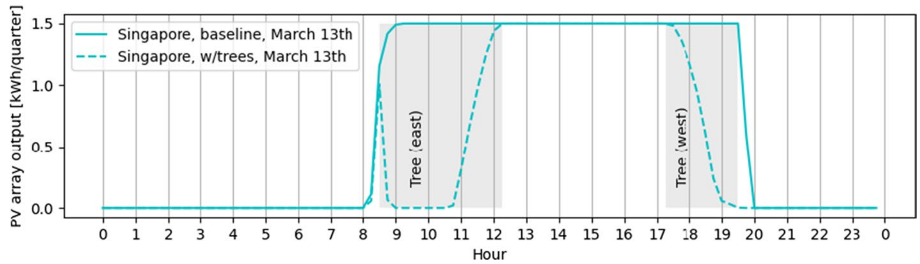

Fig. 17 The simulated quarterly PV array power output, Singapore, 13 March 2022

Fig. 18 Singapore, 13 March 2022, 08:30 local time

for all four scenarios. The yearly production is 22902.89 kWh for Odense baseline, 11950.8 kWh for Odense w/trees, 24505.23 kWh for Singapore baseline and 19214.13 kWh for Singapore w/trees. The PVs' output signal is unrealistic because the model is naive and produces the rated standard condition test power output when exposed to a direct beam. However, the patterns can be compared. The scenarios with/without trees follow the pattern of daily extraterrestrial horizontal radiation [W/m2/day] for the two positions. Therefore, UE5's 'Sun Position Calculator' plugin is a representative light source to visualize the sun. It is clearly visible that trees significantly impacted the PV array output for the whole year period. For Odense Denmark, the most significant impact is from March to May and mid-August to mid-October. For Singapore, the most significant impact is from mid-February to March and from August to mid-October.

Taking 13 March 2022 as an example, the scenario results for Singapore are shown in Fig. 17. Hours 8–9 am show a spike in production while drops in hours 9–11 am. This is explained through an illustration with collision boxes (Fig. 18) where a small opening in the east at 08:30 causes production. Still, the tree and the collision box eliminate production as sunrise until noon. The production decreases at hour 17 and is eliminated in hour 19. This is explained by the location of the west-most tree, as illustrated in Fig. 19.

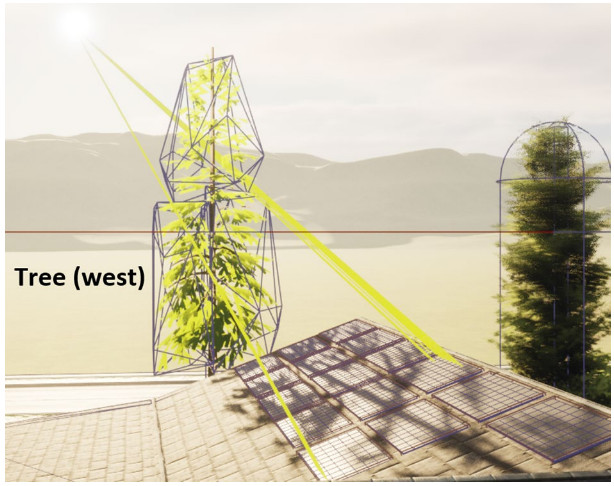

Fig. 19 Singapore, 13 March 2022, 19:30 local time

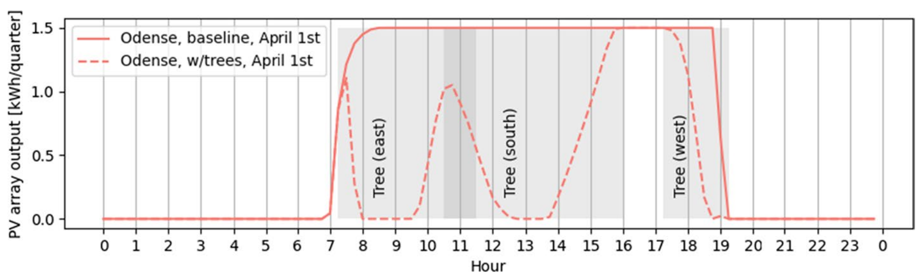

Fig. 20 The simulated quarterly PV array power output, Odense, Denmark, 1 April 2022

The scenario results for Odense Denmark on 1 April 2022 are shown in Fig. 20. 1 April 2022 was examined due to the great difference in PV output between the two scenarios for this day. The tree in the east partially eliminates production in the hours 7–12. The tree in the south partially eliminates production from hours 10–16. The overlap results from an opening between the two trees that cause a direct beam, as shown in Fig. 21.

Discussion

This paper developed a DTP of a residential rooftop PV to assess the benefits of applying UE5 for PV deployment. There are three requirements in this investigation:

The requirement 1 involves a visually correct representation of the sun in-game depending on latitude, longitude, and time of year. The plugin 'Sun Position Calculator' fulfilled this requirement, which provides an actor that can be manipulated statically and dynamically during run-time. The plugin allows a developer to parameterize the sun and thus enables simulation of scenarios over a yearly period which is used to calculate PV production estimates.

Requirement 2 is fulfilled by customising the actor PV modules and cells using only UE5. The shape and size can be customized to represent commercially available modules. Properties can be attributed to actors and components to reflect the specification of the PVs.

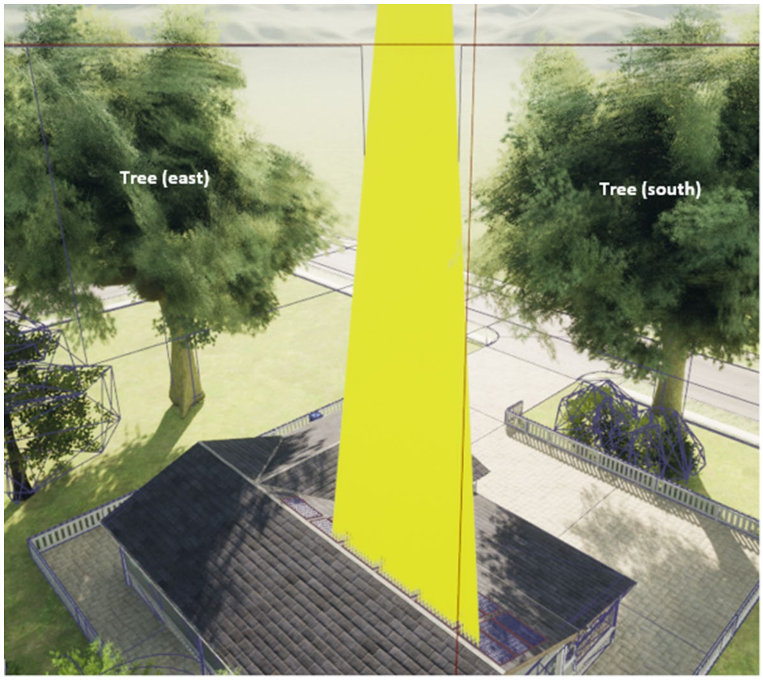

Fig. 21 Odense, Denmark, 1 April 2022, 11:00 local time

Requirement 3 is partially fulfilled. The implemented DTP could simulate the yearly PV production, but with caveats.

The results show that UE5 cannot provide direct access to information about the light collected by a given surface with the features of Lumen and the volumetric cloud system. Lumen provides tooling for developing visually stunning scenes; however, these tools are built for artists and not for practical applications that rely on non-body physics such as light in PV applications. The volumetric cloud system is not sufficient for considering cloudy and non-cloudy days. Consequently, a DT developer must implement a PV output model.

This paper implemented a naïve model that produces full capacity when cells are exposed to a direct beam, which was sufficient as DTP to assess UE5 DM capabilities in the PV domain. For real applications, it is suggested to integrate well-known PV output models, for instance, the Hay, Davies, Klucher, Reindl (HDKR) irradiance model that calculates the global radiation incident on the PVs. Such a model requires solar radiance measurements for the absolute position of interest. This data is available through various databases e.g., Baseline Surface Radiation Network (BSRN), World Radiation Data Center (WRDC), Photovoltaic Geographical Information System (PVGIS) or satellite data.

UE5 can account for objects that cast shadows on the PVs through line tracing. This mechanism can be used to track if a cell is being blocked and consequently calculate power output based only on the diffused light. Special attention must be given to objects that cast shadow. Collision boxes are applied to block the direct beam; therefore, collision boxes must be carefully modelled to represent the true blocked direct beam. Verification models should be used to determine the precision of collision boxes which is disregarded in the developed DTP. If high fidelity is a concern, trees must dynamically change behaviour during runtime depending on the year's season. For example, naked trees cast less shadow during winter. Furthermore, the suggested line tracing mechanism anticipates a clear sky view.

UE5 is built for real-time simulation, which may be sufficient for DS applications. For DM applications, mechanisms for speeding up simulation time in-game must be implemented by the DT developer. A DM executed within UE5 has a significant overhead when the model depends on the state resulting from visual rendition, which is the case with line tracing. In some cases, it may not be of interest to inspect the momentary renditions of a simulated scenario. Instead, a faster simulation model or tool might provide a better alternative. The visual rendition that comes with UE5 sacrifices execution speed for visualization. If the simulation speed is too fast, frames may be skipped leading to missing data points. Alternative simulation implementations are suggested:

Let the simulator step forward with each event tick instead of relying on the Timer.

Integrate with highly performant external simulation models and use UE5 as a visual interface to show the result.

In this paper, the 3D model of the household PV could be navigated and used to explain and visualize the decrease in PV power output at given time slots. UE5 does not provide tools for advanced plotting. Data must be exported and post-processed to visualize the simulated results. UE5 offers built-in features and some marketplace assets that enable the setup of a visually appealing environment. Ideally, UE5 should also be assessed for modelling a real-world scenario using a case study with real geometry measurements and comparing the simulated results with real measurements. The expertise required by developers are high. Currently, UE5 offers no complete solution for general scenarios with the listed requirements in the PV domain. Developers must learn UE5 and implement or integrate with PV output models through blueprints or C++. Furthermore, when the DTP transition to a DTI, any modifications to the physical system must be maintained in the digital replica. The development and maintenance costs versus benefits must be carefully considered before applying UE5 to the PV domain.

Conclusion

This paper assesses the benefits of game engines in the planning phase of PV systems. Game engines provide realistic graphics, lighting, assets, and environments that can be altered during the planning phase for presentation or simulation purposes. The developed digital twin prototype provides a PV output simulation model in 3D that considers the position of the sun, individual PV- modules and cells, and objects that block direct beam radiation. The results clearly indicate that the implemented model can estimate the impact of blocking objects of direct beam. This is indicated by the significant decrease in PV power output over a year in the chosen locations on earth.

The concepts and results presented in this paper may apply to similar game engines but requires the suggested model to be implemented and tested in such engines. The concept of digital twins is an emerging trend, and game engines have been suggested as a method of developing high-fidelity digital twins. This paper acknowledges the classification of digital twin into Digital Twin, Digital Model, and Digital Shadow.

From a PV perspective, this paper provides an early assessment of using Unreal Engine 5 to develop a digital twin prototype of a PV digital model capable of simulating various scenarios. The developed digital twin prototype stands in contrast to the existing PV applications that provide 2D graphics, schematics, plots, etc., by instead suggesting game engines as a tool not only for appealing visual environments that can be depicted from various viewpoints but also to enable scenario testing after changing the in-game environment. The contribution of this paper can be used to visualize and simulate PV application models in game engines in the planning phase.

The suggested digital twin prototype implementation is limited by being a naïve model that produces full PV capacity under a clear sky and direct beam conditions. It is recommended that future development of the implemented digital twin prototype integrates a well-known PV output model, e.g., HDKR, and real solar measurement data to provide practically usable estimates. The combination of a practical PV output model and the line tracing mechanism for activating PV cells may offer a new method of planning the deployment of PV systems. The approach should be verified through real-world case studies that integrate 3D models of existing buildings and terrain with correct physical geometry. Furthermore, the scalability of the approach can be assessed by implementing large-scale PV systems, e.g., country areas, city or utility applications. Moreover, it could be assessed whether the developed models can be modularized and packaged so other developers can benefit and integrate with the model.

Abbreviations

PV	Photovoltaic
GW	Gigawatt
TW	Terawatt
DT	Digital twin
DM	Digital model
DS	Digital shadow
DTP	Digital twin prototype
DTI	Digital Twin Instance
DTE	Digital twin environment
UI	User interface
CAD	Computer-aided design
VR	Virtual reality
IoT	Internet-of-thing
UE5	Unreal engine 5
HDKR	Hay, davies, klucher, reindl
BSRN	Baseline surface radiation network
WRDC	World radiation data center
PVGIS	Photovoltaic geographical information system

Acknowledgements
Not applicable.

About this supplement
This article has been published as part of *Energy Informatics Volume 5 Supplement 4, 2022: Proceedings of the Energy Informatics.Academy Conference 2022 (EI.A 2022)*. The full contents of the supplement are available online at https://energyinformatics.springeropen.com/articles/supplements/volume-5-supplement-4.

Author contributions
CSBC was the main contributor to the first draft writing of the manuscript. ZM was the main contributor to edit and finalize the manuscript. BNJ contribute to discussion, comments, and inputs. All authors read and approved the final manuscript.

Funding
This paper is part of the ANNEX 81 project (Project IEA EBC ANNEX 81 Data-Driven Smart Buildings, funded by EUDP Denmark, Case no.64019-0539) by the Danish funding agency (the Danish Energy Technology Development and Demonstration (EUPD) program, Denmark) and part of the ClusterSoutH2 project (project title ClusterSoutH2- Designing a PtX Ecosystem in Southern Denmark) by the European Regional Development Fund.

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Accepted: 11 October 2022
Published: 21 December 2022

References

Abdallah A, Primas M, Turcin I, Traussnigg U (2020) The potential of game development platforms for digital twins and virtual labs: case study of an energy analytics and solution lab. IFIP Adv Inf Commun Technol. https://doi.org/10.1007/978-3-030-57997-5_14

Abdallah A, Primas M, Turcin I, Traussnigg U. The potential of game development platforms for digital twins and virtual labs: case study of an energy analytics and solution lab. IFIP Advances in Information and Communication Technology. CAMPUS 02, University of Applied Sciences, Graz, 8010, Austria; p. 117–21.

Adamenko D, Kunnen S, Nagarajah A (2020) Comparative analysis of platforms for designing a digital twin. In: Innovat, Univ Duisburg Essen LDDG, editors. 3rd International Conference on Design, Simulation, Manufacturing—The Innovation Exchange (DSMIE). Kharkiv, UKRAINE: Springer International Publishing Ag; p. 3–12.

Arafet K, Berlanga R (2021) Digital twins in solar farms: an approach through time series and deep learning. Algorithms. https://doi.org/10.3390/a14050156

Autiosalo J, Vepsäläinen J, Viitala R, Tammi K (2020) A feature-based framework for structuring industrial digital twins. IEEE Access 8:1193–1208. https://doi.org/10.1109/ACCESS.2019.2950507

Barth L, Ehrat M, Fuchs R, Haarmann J (2020) Systematization of digital twins: ontology and conceptual framework. Pervasive Health. https://doi.org/10.1145/3388176.3388209

Belfadel A, Hörl S, Tapia RJ, Puchinger J. Towards a digital twin framework for adaptive last mile city logistics. 2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)2021. p. 1–6.

Billanes JD, Ma Z, Jørgensen BN (2018) The bright green hospitals case studies of hospitals' energy efficiency and flexibility in Philippines. 2018 8th International Conference on Power and Energy Systems (ICPES); p. 190–5.

Christensen K, Ma Z, Demazeau Y, Jørgensen BN (2020a) Agent-based Modeling for Optimizing CO2 Reduction in Commercial Greenhouse Production with the Implicit Demand Response. 6th IEEJ international workshop on Sensing, Actuation, Motion Control, and Optimization (SAMCON2020a). Tokyo, Japan: IEEJ Digital Library; p. 6.

Christensen K, Ma Z, Demazeau Y, Jorgensen BN. (2020b) Agent-based modeling of climate and electricity market impact on commercial greenhouse growers' demand response adoption. 2020b RIVF International Conference on Computing and Communication Technologies (RIVF). Ho Chi Minh City, Vietnam: IEEE; p. 1–7.

Christensen K, Ma Z, Værbak M, Demazeau Y, Jørgensen BN. Agent-based decision making for adoption of smart energy solutions. IV International Congress of Research in Sciences and Humanities Science and Humanities International Research Conference (SHIRCON 2019). Lima, Peru: IEEE; 2019.

D'Amico D, Ekoyuncu J, Addepalli S, Smith C, Keedwell E, Sibson J, et al. Conceptual framework of a digital twin to evaluate the degradation status of complex engineering systems. In: Mfg CMKALBE, Univ Exeter CEM, Phys Sci EEXQFDE, Babcock Int WSLWUQXE, editors. 7th CIRP Global web conference on towards shifted production value stream patterns through inference of data, models, and technology (CIRPe). Electr Network: Elsevier; 2019. p. 61–7.

Darvishi H, Ciuonzo D, Eide ER, Rossi PS (2021) Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture. IEEE Sens J 21(4):4827–4838. https://doi.org/10.1109/JSEN.2020.3029459

Delussu F, Manzione D, Meo R, Ottino G, Asare M (2022) Experiments and comparison of digital twinning of photovoltaic panels by machine learning models and a cyber-physical model in modelica. IEEE Trans Industr Inf 18(6):4018–4028. https://doi.org/10.1109/TII.2021.3108688

Demirel HO, Irshad L, Ahmed S, Tumer IY (2021) Digital twin-driven human-centered design frameworks for meeting sustainability objectives. J Comput Inf Sci Eng 21(3):10. https://doi.org/10.1115/1.4050684

Eyre JM, Lanyon-Hogg MR, Dodd TJ, Lockwood AJ, Freeman C, Scott RW. Demonstration of an industrial framework for an implementation of a process digital twin. ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE).

Fan Y, Yang J, Chen J, Hu P, Wang X, Xu J et al (2021) A digital-twin visualized architecture for flexible manufacturing system. J Manuf Syst 60:176–201. https://doi.org/10.1016/j.jmsy.2021.05.010

Felemban E, Majid ARMA, Rehman FU, Lbath A (2021) Low-cost digital twin framework for 3d modeling of homogenous urban zones. Lecture Notes Netw Syst. https://doi.org/10.1007/978-3-030-80126-7_77

Geographically Accurate Sun Positioning Tool. https://docs.unrealengine.com/5.0/en-US/geographically-accurate-sun-positioning-tool-in-unreal-engine/. Accessed 29 Jun 2022.

Granelli F, Capraro R, Lorandi M, Casari P (2021) Evaluating a digital twin of an iot resource slice: an emulation study using the ELIoT platform. IEEE Netw Lett 3(3):147–151. https://doi.org/10.1109/LNET.2021.3097556

Greis NP, Nogueira ML, Rohde W (2021) Digital twin framework for machine learning-enabled integrated production and logistics processes. In: Dolgui A, Bernard A, Lemoine D, VonCieminski G, Romero D (eds) Advances in production management systems: artificial intelligence for sustainable and resilient production systems, Apms 2021, Pt I. IFIP advances in information and communication technology. Springer International Publishing Ag, Cham, pp 218–227
Grieves M, Vickers J (2017) Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. In: Kahlen F-J, Flumerfelt S, Alves A (eds) Transdisciplinary perspectives on complex systems: new findings and approaches. Springer International Publishing, Cham, pp 85–113
Hafez W (2020) Human digital twins: two-layer machine learning architecture for intelligent human-machine collaboration. Adv Intell Syst Comput. https://doi.org/10.1007/978-3-030-39512-4_97
Howard DA, Ma Z, Veje C, Clausen A, Aaslyng JM, Jørgensen BN (2021) Greenhouse industry 4.0—digital twin technology for commercial greenhouses. Energy Inform. https://doi.org/10.1186/s42162-021-00161-9
Howard DA, Ma Z, Jørgensen BN (2021) Digital twin framework for energy efficient greenhouse industry 40. In: Novais P, Vercelli G, Larriba-Pey JL, Herrera F, Chamoso P (eds) Ambient intelligence—software and applications. Springer International Publishing, Cham, pp 293–297
Howard D, Ma Z, Engvang J, Hagenau M, Jørgensen K, Olesen J, et al. Optimization of energy flexibility in cooling process for brewery fermentation with multi-agent simulation. 6th IEEJ International Workshop on Sensing, Actuation, Motion Control, and Optimization. Shibaura Institute of Technology, Tokyo, Japan.
Hyeong-su K, Jin-Woo K, Yun S, Kim W. (2019) A novel wildfire digital-twin framework using interactive wildfire spread simulator. 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), p. 636–8.
Karakra A, Lamine E, Fontanili F, Lamothe J. HospiT'Win: A digital twin framework for patients' pathways real-time monitoring and hospital organizational resilience capacity enhancement. 9th International Workshop on Innovative Simulation for Health Care, IWISH 2020. p. 62–71.
Khan A, Shahid F, Maple C, Ahmad A, Jeon G (2020) Towards smart manufacturing using spiral digital twin framework and twinchain. IEEE Trans Industr Inf. https://doi.org/10.1109/TII.2020.3047840
Kong LCW, Harper S, Mitchell D, Blanche J, Lim T, Flynn D (2020) Interactive digital twins framework for asset management through internet. 2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT). p. 1–7.
Kritzinger W, Karner M, Traar G, Henjes J, Sihn W (2018) Digital twin in manufacturing: a categorical literature review and classification. IFAC-PapersOnLine 51(11):1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474
La Russa FM, Santagati C (2020) Historical sentient—Building information model: a digital twin for the management of museum collections in historical architectures. ISPRS Arch. 43:755–762
Landahl J, Panarotto M, Johannesson H, Isaksson O, Lööf J. Towards adopting digital twins to support design reuse during platform concept development. Proceedings of NordDesign: Design in the Era of Digitalization, NordDesign 2018.
Lee D, Lee SH, Masoud N, Krishnan MS, Li VC (2021) Integrated digital twin and blockchain framework to support accountable information sharing in construction projects. Autom Constr 127:9. https://doi.org/10.1016/j.autcon.2021.103688
Lehner D, Pfeiffer J, Tinsel EF, Strljic MM, Sint S, Vierhauser M et al (2021a) Digital twin platforms: requirements, capabilities, and future prospects. IEEE Softw. https://doi.org/10.1109/MS.2021a.3133795
Lehner D, Sint S, Vierhauser M, Narzt W, Wimmer M. (2021b) AML4DT: A model-driven framework for developing and maintaining digital twins with automationML. 2021b 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA); p. 1–8.
Leinonen E, Firouzian A, Partanen C, Pulli P. (2019) Visual validation services with time coordination for senior citizens social events—OldBirds digital twin platform. 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC). p. 1–7.
Leskovsky R, Kucera E, Haffner O, Rosinova D. Proposal of digital twin platform based on 3d rendering and iiot principles using virtual/augmented reality. Proceedings of the 30th International Conference on Cybernetics and Informatics, K and I 2020.
Li Y, Chang DF, Gao YP, Zou Y, Bao CT (2021a) Automated container terminal production operation and optimization via an adaboost-based digital twin framework. J Adv Transp 2021:16. https://doi.org/10.1155/2021/1936764
Li H, Lu JZ, Zheng XC, Wang GX, Kiritsis D (2021b) Supporting digital twin integration using semantic modeling and high-level architecture. In: Dolgui A, Bernard A, Lemoine D, VonCieminski G, Romero D (eds) Advances in production management systems: artificial intelligence for sustainable and resilient production systems, apms 2021b, Pt Iv. IFIP advances in information and communication technology. Springer International Publishing Ag, Cham, pp 228–236
Lin WD, Low MYH (2020) Concept design of a system architecture for a manufacturing cyber-physical digital twin system. 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM); p. 1320–4.
Liu J, Lu X, Zhou Y, Cui J, Wang S, Zhao Z (2021) Design of photovoltaic power station intelligent operation and maintenance system based on digital twin. IEEE; p. 206–11.
Ma Z, Santos AQ, Gamborg F, Nielsen JF, Johannesen JM, Jensen MDH, et al (2018) Solutions for Remote Island Microgrids: Discussion and analysis of Indonesia's remote island energy system. The International Conference on Innovative Smart Grid Technologies (IEEE PES ISGT Asia 2018). Singapore: IEEE; 2018.
Ma Z, Broe M, Fischer A, Sørensen TB, Frederiksen MV, Jøergensen BN (2019a) Ecosystem thinking: creating microgrid solutions for reliable power supply in India's power system. 2019a 1st Global Power, Energy and Communication Conference (GPECOM); p. 392–7.
Ma Z, Værbak M, Rasmussen RK, Jøergensen BN (2019b) Distributed energy resource adoption for campus microgrid. IEEE 17th International Conference on Industrial Informatics (INDIN); p. 1065–70.
Marmolejo-Saucedo JA (2021) Digital twin framework for large-scale optimization problems in supply chains: a case of packing problem. Mobile Netw Appl. https://doi.org/10.1007/s11036-021-01856-9
Meta I, Serra-Burriel F, Carrasco-Jiménez JC, Cucchietti FM, Diví-Cuesta C, García Calatrava C et al (2021) The camp nou stadium as a testbed for city physiology: a modular framework for urban digital twins. Complexity. https://doi.org/10.1155/2021/9731180
Michael J, Wortmann A (2021) Towards development platforms for digital twins: a model-driven low-code approach. IFIP Adv Inf Commun Technol. https://doi.org/10.1007/978-3-030-85874-2_35

Min Q, Lu Y, Liu Z, Su C, Wang B (2019) Machine learning based digital twin framework for production optimization in petrochemical industry. Int J Inf Manage 49:502–519. https://doi.org/10.1016/j.ijinfomgt.2019.05.020
Minerva R, Crespi N (2021) Digital twins: properties, software frameworks, and application scenarios. IT Prof 23(1):51–55. https://doi.org/10.1109/MITP.2020.2982896
Mourtzis D, Angelopoulos J, Panopoulos N, Kardamakis D (2021) A smart IoT platform for oncology patient diagnosis based on ai: towards the human digital twin. Proc CIRP 104:1686–1691
Negri E, Pandhare V, Cattaneo L, Singh J, Macchi M, Lee J (2021) Field-synchronized digital twin framework for production scheduling with uncertainty. J Intell Manuf 32(4):1207–1228. https://doi.org/10.1007/s10845-020-01685-9
Negri E, Ardakani HD, Cattaneo L, Singh J, Macchi M, Lee J. A Digital Twin-based scheduling framework including Equipment Health Index and Genetic Algorithms. In: Control IFACTCEI, Networking, Politecn Milan DME, Ind Engn IMI, Univ Cincinnati UCRCIMSNSFICOHUSA, editors. 13th International-Federation-of-Automatic-Control (IFAC) Workshop on Intelligent Manufacturing Systems (IMS). Oshawa, CANADA: Elsevier; 2019. p. 43–8.
Negrin DAM, Cleophas L, Brand MVD (2021) Using ptolemy II as a framework for virtual entity integration and orchestration in digital twins. 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), p. 233–6.
Park KT, Son YH, Noh SD (2020) The architectural framework of a cyber physical logistics system for digital-twin-based supply chain control. Int J Prod Res. https://doi.org/10.1080/00207543.2020.1788738
Perabo F, Park D, Zadeh MK, Smogeli O, Jamt L (2020) Digital twin modelling of ship power and propulsion systems: application of the open simulation platform (OSP). IEEE International Symposium on Industrial Electronics p. 1265–70.
Pereira JG, Ellman A (2020) FROM CAD to PHYSICS-BASED DIGITAL TWIN: FRAMEWORK for REAL-TIME SIMULATION of VIRTUAL PROTOTYPES. Proceedings of the Design Society: DESIGN Conference p. 335–44.
Piascik B, Vickers J, Lowry D, Scotti S, Stewart J, Calomino A (2010) Materials, structures, mechanical systems, and manufacturing roadmap. National Aeronautics and Space Administration (NASA), Washington
Piroumian V (2021) Digital twins: universal interoperability for the digital age. Computer 54(1):61–69. https://doi.org/10.1109/MC.2020.3032148
Raes L, Michiels P, Adolphi T, Tampere C, Dalianis T, Mcaleer S et al (2021) DUET: a framework for building secure and trusted digital twins of smart cities. IEEE Internet Comput. https://doi.org/10.1109/MIC.2021.3060962
Razo DEG, Müller B, Madsen H, Wittwer C (2020) A genetic algorithm approach as a self-learning and optimization tool for PV power simulation and digital twinning. Energies 13(24):6712. https://doi.org/10.3390/en13246712
Rodríguez-Aguilar R, Marmolejo-Saucedo JA (2020) Conceptual framework of digital health public emergency system: digital twins and multiparadigm simulation. EAI Endorsed Trans Pervasive Health Technol. https://doi.org/10.4108/eai.13-7-2018.164261
Schmela M (2022) Global market outlook for solar power 2022–2026. SolarPower Europe
Shah K, Prabhakar TV, Sarweshkumar CR, Abhishek SV (2021) Construction of a digital twin framework using free and open-source software programs. IEEE Internet Comput. https://doi.org/10.1109/MIC.2021.3051798
Shao GD, Helu M (2020) Framework for a digital twin in manufacturing: scope and requirements. Manufacturing Letters 24:105–107. https://doi.org/10.1016/j.mfglet.2020.04.004
Sharotry A, Jimenez JA, Wierschem D, Mediavilla FAM, Koldenhoven RM, Valles D, et al. (2020) A digital twin framework for real-time analysis and feedback of repetitive work in the manual material handling industry. Proceedings of the Winter Simulation Conference. Orlando, Florida: IEEE Press; p. 2637–48.
Værbak M, Ma Z, Christensen K, Demazeau Y, Jørgensen BN (2019) Agent-based modelling of demand-side flexibility adoption in reservoir pumping. 2019 IEEE Sciences and Humanities International Research Conference (SHIRCON) p. 1–4.
Viola J, Chen Y (2020) Digital Twin enabled smart control engineering as an industrial ai: a new framework and case study. 2020 2nd International Conference on Industrial Artificial Intelligence (IAI). p. 1–6.
Volkov I, Radchenko G, Tchernykh A (2021) Digital twins, internet of things and mobile medicine: a review of current platforms to support smart healthcare. Program Comput Softw 47(8):578–590. https://doi.org/10.1134/S0361768821080284
Vrabič R, Erkoyuncu JA, Farsi M, Ariansyah D (2021) An intelligent agent-based architecture for resilient digital twins in manufacturing. CIRP Ann 70(1):349–352. https://doi.org/10.1016/j.cirp.2021.04.049
Wu Z, Li J (2021) A framework of dynamic data driven digital twin for complex engineering products: the example of aircraft engine health management. Proc Manufact. 55:139–146
Xu WJ, Cui J, Li L, Yao BT, Tian SS, Zhou ZD (2021) Digital twin-based industrial cloud robotics: framework, control approach and implementation. J Manuf Syst 58:196–209. https://doi.org/10.1016/j.jmsy.2020.07.013
Zhang X, Li R, Wang Y, Manandhar U (2021) Digital twin real-time hybrid simulation platform for engineering education in renewable energy. 2021 31st Australasian Universities Power Engineering Conference (AUPEC); p. 1–6.
Zhang J, Li P, Luo L (2021) Digital twin-based smart manufacturing cell: application case, system architecture and implementation. J Phys Conf Series. 1884(1):012017
Zhang T, Sun S (2021) An exploratory multi-scale framework to reservoir digital twin. Adv Geo-Energy Res 5(3):239–251. https://doi.org/10.46690/ager.2021.03.02
Zhifeng LIU, Yueze Z, Caixia Z, Jun YAN, Shiyao GUO (2021) Real-time workshop digital twin scheduling platform for discrete manufacturing. J Phys Conf Series. https://doi.org/10.1088/1742-6596/1884/1/012006/meta
Zohdi TI (2020) A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments. Comput Methods Appl Mech Eng 363:19. https://doi.org/10.1016/j.cma.2020.112907
Zohdi TI (2021a) A digital twin framework for machine learning optimization of aerial fire fighting and pilot safety. Comput Methods Appl Mech Eng 373:17. https://doi.org/10.1016/j.cma.2020.113446
Zohdi TI (2021b) A digital-twin and machine-learning framework for the design of multiobjective agrophotovoltaic solar farms. Comput Mech 68:357–370. https://doi.org/10.1007/s00466-021-02035-z

Publisher's Note