[Array 14 (2022) 100176](https://doi.org/10.1016/j.array.2022.100176)

|  |  |  |
| --- | --- | --- |
|  | Contents lists available at [ScienceDirect](http://www.elsevier.com/locate/array) |  |
| Array |
| journal homepage: [www.elsevier.com/locate/array](http://www.elsevier.com/locate/array) |
|  | | |

Industrial digitalization in the industry 4.0 era: Classification, reuse and authoring of digital models on Digital Twin platforms

Valentina Zambranoa,∗, Johannes Mueller-Roemerb, Michael Sandbergc, Prasad Talasilac, Davide Zanind, Peter Gorm Larsenc, Elke Loeschnere, Wolfgang Thronickee, Dario Pietraroiaf, Giuseppe Landolfid, Alessandro Fontanad, Manuel Laspalasa, Jibinraj Antonyg, Valerie Poserg, Tamas Kissh, Simon Bergweilerg, Sebastian Pena Sernai, Salvador Izquierdoa, Ismael Viejoa, Asier Juana, Francisco Serranoa, André Storkb

a *Instituto Tecnológico de Aragón – ITAINNOVA, C/ María de Luna 7-8, 50018 Zaragoza, Spain*   
b *Fraunhofer Institute for Computer Graphics Research IGD, Fraunhoferstraße 5, 64283 Darmstadt, Germany*   
c *Aarhus University, Nordre Ringgade 1, 8000 Aarhus, Denmark*   
d *Scuola Universitaria Professionale della Svizzera Italiana – SUPSI, Via Pobiette 11, 6928 Manno, Switzerland*   
e *Atos Information Technology GmbH, Otto-Hahn-Ring 6, 81739 Munich, Germany*   
f *Technology Transfer System – TTS, Via Francesco d’Ovidio 3, 20131 Milano, Italy*   
g *Deutsches Forschungszentrum Für Künstliche Intelligenz – DFKI GmbH, Trippstadter Str. 122, 67663 Kaiserslautern, Germany* h *University of Westminster, Centre for Parallel Computing, New Cavendish Street, W1W 6UW London, United Kingdom* i *Clesgo GmbH, Seyfferstraße 34, 70197 Stuttgart, Germany*

|  |  |  |
| --- | --- | --- |
| A R T I C L E | I N F O | A B S T R A C T |
| *Keywords:*  Digital Twin  Cyber–Physical System  Industry 4.0  Reusable models | | Digital Twins (DTs) are real-time digital models that allow for self-diagnosis, self-optimization and self-configuration without the need for human input or intervention. While DTs are a central aspect of the ongoing fourth industrial revolution (I4.0), this leap forward may be reserved for the established, large-cap companies since the adoption of digital technologies among Small and Medium-size Enterprises (SMEs) is still modest. The aim of the H2020 European Project "DIGITbrain" is to support a modular construction of DTs by reusing their fundamental building blocks, i.e., the *Models* that describe the behavior of the DT, their associated *Algorithms* and the *Data* required for the evaluation. By offering these building blocks as a service via a DT Platform (a Digital Twin Environment), the technical barriers among SMEs to adopt these technologies are lowered. This paper describes how digital models can be classified, reused and authored on such DT Platforms. Through experimental analyses of three industrial cases, the paper exemplifies how DTs are employed in relation to product assembly of agricultural robots, polymer injection molding, as well as laser-cutting and sheet-metal forming of aluminum. |

**1. Introduction**

A successful manufacturing company must be agile, innovative, and highly efficient. Not only do companies today face fierce competition from globalization, but product requirements constantly increase due to new legislation, regulations, and customer expectations. Therefore, manufacturing companies already exploit digitalization techniques to be successful in the competitive market. This is the basis of the ongoing fourth industrial revolution (i.e., I4.0), which is already bringing a paradigm shift to manufacturing engineering.

I4.0 builds on the foundation of *Internet of Things* (IoT) technology, where sensors and software are embedded in devices (i.e., cyber–physical systems) to exploit different aspects of computerization. As such, a central aspect of I4.0 is the deployment of the so-called Dig-ital Twins (DTs) [1] that ultimately act as real-time digital models and allow for self-diagnosis, self-optimization and self-configuration without the need for human input or intervention. There is, however, data that indicate that the leap forward offered by I4.0 might be reserved for the established, large-cap companies. For example, the Digital Transformation Scoreboard in EU signals that the adoption of digital technologies among SMEs and mid-caps [2] is below 10%, which

∗ Corresponding au[thor.](mailto:vzambrano@itainnova.es)

*E-mail addresses:* [vzambrano@itainnova.es](mailto:vzambrano@itainnova.es) (V. Zambrano), [prasad.talasila@ece.au.dk](mailto:prasad.talasila@ece.au.dk) (P. Talasila).

<https://doi.org/10.1016/j.array.2022.100176>  
[Received 25 April 2022; Accepted 25 April 2](https://doi.org/10.1016/j.array.2022.100176)022   
Available online 10 May 2022

2590-0056/© 2022 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

*V. Zambrano et al.*  *Array 14 (2022) 100176*

poses a significant barrier if I4.0 technologies are to be adopted across the entire manufacturing sector.

DTs are becoming increasingly prominent in the manufacturing industry. DTs are capable of optimizing processes and products prior to their execution thanks to their ability of high-fidelity forecasting. In manufacturing industries, DTs can be virtual representations of differ-ent systems, e.g., machinery, production lines, products, or any other operation or service related to the manufacturing process. Therefore, DTs can be introduced at any time in the manufacturing process, hence their usefulness is not limited to pre-production planning and design, where prediction and optimization of the product or production lines apply, but they can also be used for maintenance, market analysis, etc. [3].

One way to allow SMEs and mid-caps access to I4.0 technologies is to offer the services via a platform, which is sometimes referred to as a *Digital Twin Environment* [4]. Not only does this lower the investment and infrastructure requirements of companies, but it also gives non-technical staff access to key technologies with only limited training. A Digital Twin Environment can utilize collected data to let models and algorithms remotely steer and optimize products and processes ac-cording to the operating conditions. As such, manufacturing companies can now outsource both expertise and parts of the supply chain [5] to service providers, which is referred to as Manufacturing as a Service (MaaS) [6–10].

To provide a MaaS platform, several building blocks are needed, many of which have already been explored, among others, in previously funded European H2020 projects. A non-exhaustive list of these EU projects includes: digital services marketplaces for manufacturing com-panies (CloudiFacturing [11], MANUSQUARE [12]); deployments of smart applications in open-source IoT platforms (FIWARE [12]); cloud orchestration engines (COLA [13,14]); as well as basic research into cyber–physical systems (INTO-CPS [15]) and simulation and forecast-ing technologies (MAYA [16]). The natural step from present state of the art is to develop a complete set of solutions that further extends the concept of DTs and enable manufacturing companies to tap into the full potential of I4.0. This is the goal of the European H2020 project DIGITbrain and this paper presents the project’s novel approach how models for Digital Twins can be made reusable based on the DIGITbrain technology [17].

In order to provide a digital integrated platform, both software and hardware components need to be considered. However, one of the core technologies of the DIGITbrain project is the device-agnostic software-based verification mechanism by the implementation of a lightweight cryptographic library. This paper strictly focuses on Models, as a sepa-rate and reusable asset in DIGITbrain.

This paper is organized as follows: Section 2 provides background for the paper by reviewing related work; Section 3 identifies and reviews a classification of models that may be used on MaaS platforms for evaluation, including the behavior of models, organization, and interaction of coupled models, embedded models, and stateful and stateless models; Section 4 presents the proposed model characteri-zation and metadata structure within DIGITbrain; Section 5 explains how models can be authored; Section 6 lists some of the most relevant results obtained during the first year of DIGITbrain Project leading to enhanced model reusability; and finally, Section 7 concludes the findings of the paper.

**2. Background and related work**

The use of DTs dates more than 50 years back to the Apollo 13 program, where NASA had designed physical simulators to mirror the conditions that astronauts would experience in the spacecraft during spaceflight [18]. While no formal, generic definition followed its ini-tial introduction, the consensus now characterizes a DT as a digital model with near real-time, bidirectional communication to the physical system [19]. Accordingly, models with one-way communication are

2

*V. Zambrano et al.*  *Array 14 (2022) 100176*

time-consuming process. Each DT is typically constructed individually, without reusing previous results and existing building blocks. The work described in this paper is the first step to overcome this obstacle by categorizing and describing models with a rich set of metadata to enable their reusability when building DTs.

**3. Types of models**

The DIGITbrain Project includes and supports several kinds of mod-els. A model in DIGITbrain is an asset that contains the knowledge related to a specific industrial product instance (i.e., a concrete man-ufacturing machine or production line), which can hence describe and forecast the behavior of such an instance when specific operating con-ditions are given (n.b., the process of forecasting a system’s behavior according to specific operating conditions is also known as model evaluation). In this section, we detail the different model types.

*3.1. Co-simulation models*

Models related to Cyber–Physical Systems (CPSs) are often difficult to develop, due to the large variety of sub-systems (e.g., networks, control algorithms, mechanical components, electrical circuits, sen-sors) and components with different formalisms. Since sub-systems and components can be reused in different scenarios and applications, it becomes very helpful to be able to model them separately, instead of creating a single monolithic model that includes all of them. For this purpose, co-simulation is a very useful approach for coupling together the different parts of a whole system.

The sub-systems and components are created by model developers and exported as co-simulation units [41]. In order to perform co-simulation of a complete system, these co-simulation units need a standard way of interacting with each other. Since manufacturing systems are typically produced by a combination of heterogeneous components, these components are usually supplied by several legal entities; hence protection of the intellectual property for the underlying model is needed.

The Functional Mockup Interface (FMI) [42] is a widely used so-lution for the above described problem where the orchestration of combining differing simulation units is made by independent orchestra-tion engines [43]. FMI is a cross-platform and open-source standard to exchange models and perform co-simulation. Components that conform to the FMI standard are called Functional Mockup Units (FMUs). An FMU is distributed as an archive with the file extension *.fmu*. An FMU contains the following:

• an XML-file with metadata, definitions of all the variables inside the FMU, and desired outputs  
• implementation in source and/or binary form, specifically:

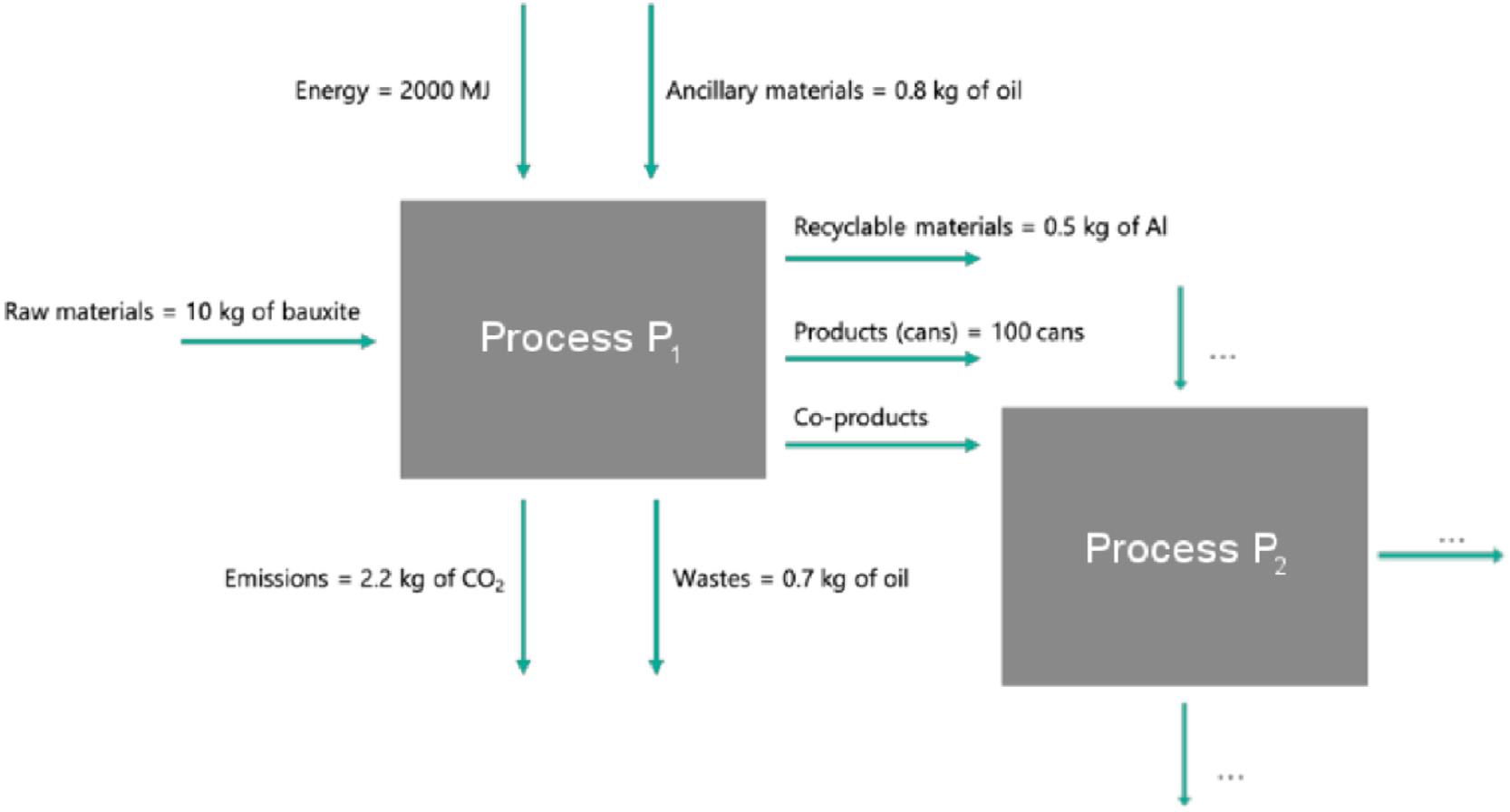
**–** binaries: this directory contains the executable files of the FMU and can also contain shared library executable code for different OS (Operating System) platforms  
**–** resources (optional): the contents of this directory can be used by the FMU during execution time  
**–** sources (optional): the source code of the FMU, compiled to produce the shared libraries placed in the binaries directory

• additional Data, such as documentation

It is important to remark that the model itself inside the FMU may be distributed as binary and hence can be exchanged as *black boxes*, which is also beneficial to protect the intellectual property of the source code.

3

*V. Zambrano et al.*  *Array 14 (2022) 100176*



**Fig. 1.** Example of a SoS characterization from an LCA perspective.

where M is the ROM’s approximation order, so that the first addend corresponds to a first approximation of the system, while following terms are corrections to it, being *𝛼𝑚, 𝑚* = 1*,* … *, 𝑀* weighting coefficients with generally decreasing values.

CAELIATMROMs, for instance, can be easily obtained from input data files such as *.txt* or *.csv* that typically include data points where both inputs and output values are provided, similarly to supervised ML algorithms [57]. These ROM models can be easily embedded and managed by virtually any kind of environment, e.g., desktop applica-tions or web interfaces, where users can provide keyboard inputs and navigate to obtain real-time results. The computed ROM model file can be straightforwardly embedded in different platforms as a *.txt* file that includes all the necessary information and parameters to compute the system’s response in real-time.

*3.4. System of Systems models*

System of Systems (SoS) models can be based on both structural and behavioral models. One example of a SoS model, from a structural point of view, is the Life Cycle Assessment (LCA) tool [58] model applied in a manufacturing domain. LCA is a well-acknowledged methodology for analyzing the environmental impacts of manufacturing processes along their entire life cycle. LCA assessment leverages on the use of well-founded background data that enable users to personalize their operation information so to calculate the environmental indicators that represent impacts related to a specific process. This process is sup-ported by the LCA Process Templates (PT) tool which creates process characterization by formalizing the Life Cycle Inventory (LCI) descrip-tion, where for each specific process inputs and outputs are identified and quantified. LCI considers resources (i.e., inputs) coming from the ecosphere (e.g., raw material, water) or from another technosphere (e.g., ancillary material such as lubricating oil) and energy of various types. LCI assesses emissions (i.e., outputs), such as waste, products and co-products. LCI data represent the variables belonging to the LCA model and are retrieved directly from the production line or collected manually through IoT devices. Fig. 1 shows an example of LCA process characterization. The figure shows how LCI output of a certain process can become the input for another process in a SoS model.

From a LCA point of view, an output of a specific process (i.e., a system) can be formalized as an input to another process. Since a process is considered the fundamental unit for the LCA evaluation, a complex system might be represented by a production machine (as

4

*V. Zambrano et al.*  *Array 14 (2022) 100176*

Distinguishing between stateful and stateless models is particularly important for the correct evaluation of models. On the one hand, while stateless models can be reconstructed at any time, stateful models and their evaluating algorithms must remain instantiated on a given compute node. Alternatively, the algorithm must support serializing its state and restoring it on a different machine. On the other hand, keeping stateless models instantiated is not a requirement, although it might be beneficial to avoid instantiation overhead. In this context, it is important to consider the cost of on-demand instantiation of a model with a given algorithm, as well as (de)serialization costs for stateful models. If the cost of on-demand instantiation is less than the benefit derived from continuous model evaluation, then on-demand instantiation might be a better choice.

**4. Reusable models for DIGITbrain**

|  |  |  |
| --- | --- | --- |
| **Table 1**  Selection of model metadata related to model definition. | | |
| Key | Type | Description |
| ID | UUID | SemVer ID of the model.  Name of the model.  Model version.  Licensing model chosen from a fixed set of known licenses. |
| name | string |
| version | SemVer2 |
| license | string |
| provider | enum | Provider name: Institution or Person. *(optional)* Dictionary with keys being phone, email, address. |
| provider\_contacts | obj |
| AuthTool | obj | Authoring tool used to create the model. Model type, e.g., ML, LCA, 3D FEM,  CFD, system simulation, discrete event simulation, or co-simulation; any  algorithm that supports the given type can be used to evaluate this model. |
| type | enum |

|  |  |  |  |
| --- | --- | --- | --- |
| *4.1. Reuse in the DIGITbrain ecosystem* | fidelity | number | *(optional)* Error of the model’s prediction. |

The key to efficiently create DTs lies in the reuse of existing arti-facts. The reuse of software artifacts is a well-established concept in software engineering [59]. The DIGITbrain ecosystem enables reuse by decomposing DTs into their constituting parts, i.e., data, models, and al-gorithms. Metadata describing these assets and their physical locations in external repositories are stored in the Platform, enabling users to create DTs by composing the parts into a Data-Model-Algorithm (DMA) tuple.

Focusing on models, the core problem is not the availability, but the identification, occasional adaptation, and composition of models with suitable data and algorithms to create DTs (i.e., DMA Tuples). In DIGITbrain a model classification has been developed to allow a model developer creating a meta-description, which ensures discoverability and composability of models with algorithms by DT experts after publication.

The major challenge tackled by DIGITbrain is to present the charac-terization in such a way that the model user can conveniently find the right model for efficient reuse. The core service responsible for enabling reuse is the DIGITbrain Asset Metadata Registry that stores a rich set of metadata about all DIGITbrain assets, including models. The actual models are only referenced from this Registry and stored in external model repositories (the project set up a sample model repository for demonstration purposes). The access service relies on the metadata descriptions to provide filtering, search functionality and to assure that the selected model is fed into the chosen algorithm at execution time. Within this context, the main classification characteristics proposed in DIGITbrain are the various aspects of the model, for example mod-eling language, model inputs and outputs, and the structure and con-struction of the model. Moreover, further characteristics are taken into account, related to the evaluation of the model by the selected algorithm, such as its storage requirements during execution, and the information about the model’s fidelity in its range of validity.

For this purpose, generalized metadata description tables were cre-ated (see Tables 1 to 3), to be filled with the selected characteristics of each model. The tables, represented as key–value pairs, are described in the following sections.

*4.2. Model metadata description*

The process of publishing a model on the DIGITbrain Platform requires providing all information needed for the model to be evaluated with a compatible algorithm. Once the model information has been collected, it can be published on the DIGITbrain Platform, using the dedicated DIGITbrain publishing interface.

As mentioned earlier, in order to facilitate data collection from different types of models, metadata specifications for models have been defined as a common structure to be used within the DIGITbrain Plat-form and are stored as key–value pairs in a relational database. Model

5

|  |  |  |  |
| --- | --- | --- | --- |
| *V. Zambrano et al.* | *Array 14 (2022) 100176* | | |
| **Table 2**  Selection of model metadata related to model parametrization. | | | |
| Key | Type | Description | |
| in\_slots | array[object] | | Input values and/or parameters for the model. The objects in this array contain input’s or parameter’s: unique key, name, number of dimensions, units (i.e., a human-readable name, SI – International System of Units – exponents in a [m, s, mol, A, K, cd, kg] format, a scale offset if needed and a scale’s order of magnitude), default value if available, ranges (i.e., minima and maxima) and a description |
| outputs | array[object] | | Model-specific outputs. Structured analogously to *in\_slots*. |
| cosim\_solver\_info | object *(optional)* | | Co-simulation models bundle binary solvers. Therefore, execution information such as operating system, CPU and GPU architecture, etc. are required in that case (see Tables 4 to 6). |

|  |  |  |
| --- | --- | --- |
| **Table 3**  Selection of model metadata related to model publication. | | |
| Key | Type | Description |
| description | CommonMark | Human-readable description of the  model, e.g., version, scope (simulation, control, etc.). Provided as CommonMark Markdown for rendering as a web page. |
| **Table 4**  Cosim\_solver\_info Dependant FMUs metadata *(optional)*. | | |
| Key | Type | Description |
| dependencies | array[URI] | Dependant FMUs for co-simulation. |
| **Table 5**  Cosim\_solver\_info OS requirements metadata *(optional)*. | | |
| Key | Type | Description |
| osArch | enum | OS architecture type (e.g., x86\_64). OS type (e.g., Windows, Linux).  OS distribution (e.g., Ubuntu, Fedora). Version of the OS. |
| osType | enum |
| osDistribution | enum |
| osVersion | SemVer2 |

into executable DMA Tuples becomes possible at run-time. Obviously, the compatibility of the combined assets still needs to be checked and assured, either by the human composer (as in the current version of the DIGITbrain Platform) or via an ontology-based automated approach (which may be considered for the future). However, from the pub-lished metadata the DIGITbrain Platform is capable of generating the executable ‘‘artifact’’ automatically by combining the model with the algorithm that evaluates it and the data that is required as input for the calculation, without further human intervention.

Please note that while the above-mentioned composition and auto-matic code generation are important and interesting topics, these are out of the scope of this paper as we only concentrate on models and their reusability.

**5. Authoring of DIGITbrain models**

There are several authoring tools for generating, importing, man-aging, evaluating, and exporting various models. In this section the authoring tools related to DIGITbrain models detailed in Section 3 are described. Please note that such authoring tools are outside the DIGITbrain Platform. With such decision DIGITbrain does not limit the choice of authoring tools that can be used to generate models. Any authoring tool can be applied, and the generated models can be registered in a uniform way with the Platform, as described in Section 4. As a consequence, the authoring tools detailed in this section are examples only. Other authoring tools can also be used freely in relation to the DIGITbrain approach.

*5.1. Authoring tools for co-simulation models*

Many modeling and simulation tools are used to author functional mockup unit (FMUs) – co-simulation units – conforming to the func-tional mockup interface (FMI) standard; over 100 of these authoring

6

|  |  |  |
| --- | --- | --- |
| *V. Zambrano et al.* |  | *Array 14 (2022) 100176* |
| **Table 6**  Cosim\_solver\_info Hardware requirements metadata *(optional)*. |
| Key | Type | Description |
| recommendedNumberOfGPUCores | number | Recommended number of GPU cores.  Minimum required number of GPU cores. Recommended GPU memory.  Minimum required GPU memory.  Recommended Memory.  Minimum required memory.  Recommended number of CPU cores.  Minimum required number of CPU cores.  Required amount of disk space in GB. |
| minimumNumberOfGPUCores | number |
| recommendedGPURAM | number |
| minimumGPURAM | number |
| recommendedRAM | number |
| minimumRAM | number |
| recommendedCPUs | number |
| minimumCPUs | number |
| requiredDiskSpace | number |

*5.3. Authoring tools for Reduced Order Models*

As described in Section 3, ROM is an umbrella of methods that can be handful in different situations where a reduction in terms of system variables and their relations should be taken into consideration. The most common scenarios for ROM application are the followings:

• a large amount of data is to be analyzed,  
• the physics behind the system under investigation remains uncer- tain,  
• the system’s equations are well known, but the solution is not trivial and/or its computation is highly time- and resource con- suming.

To support such scenarios, the CAELIA™ tool was developed at ITAINNOVA for ROM generation and management, using TRD tech-nique. CAELIA™ can be used for different manufacturing processes, such as injection molding, rubber extrusion, hot stamping, and laser welding. CAELIA™ consists of a set of libraries, where models are gen-erated using Twinkle library [56]. Since TRD-based ROM models are based on data (please, refer to Section 3) that can be obtained through experiments, simulations or by mathematically solving equations (if known), a careful Design of Experiment (DoE) must be performed beforehand. Hence, this authoring tool also includes a library for au-tomatic DoE with an enhanced space coverage. Additionally, CAELIA™includes different optimization algorithms and several visualization interfaces. All these libraries and tools can be combined and used for enhancing ROM computation and management.

CAELIA™’s ROMs are a valuable tool, not only capable of forecasting unexplored behaviors of the system, but also useful for optimization, i.e., for finding the system’s operating conditions that correspond to a desired output.

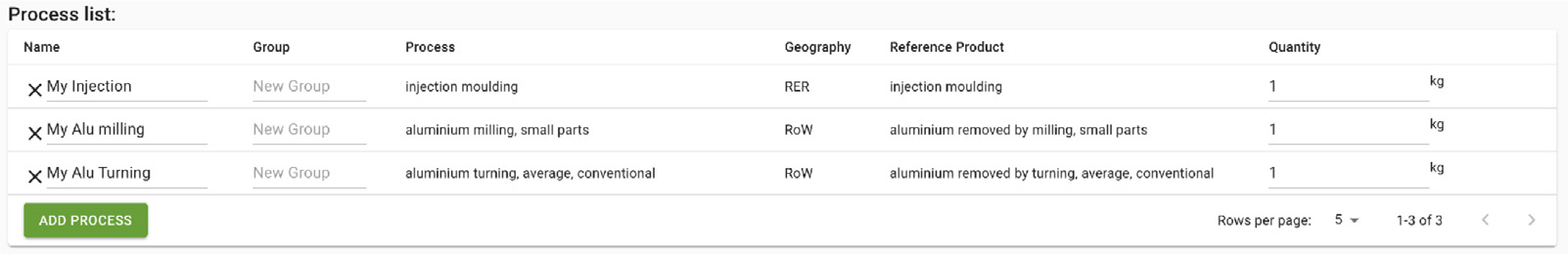
The CAELIA™ toolset has been already tested against a large variety of cases, such as fluid- and thermodynamics, friction modeling, etc. In Section 6 an injection application CAELIA™ will be detailed.

*5.4. Authoring tools for System of Systems models*

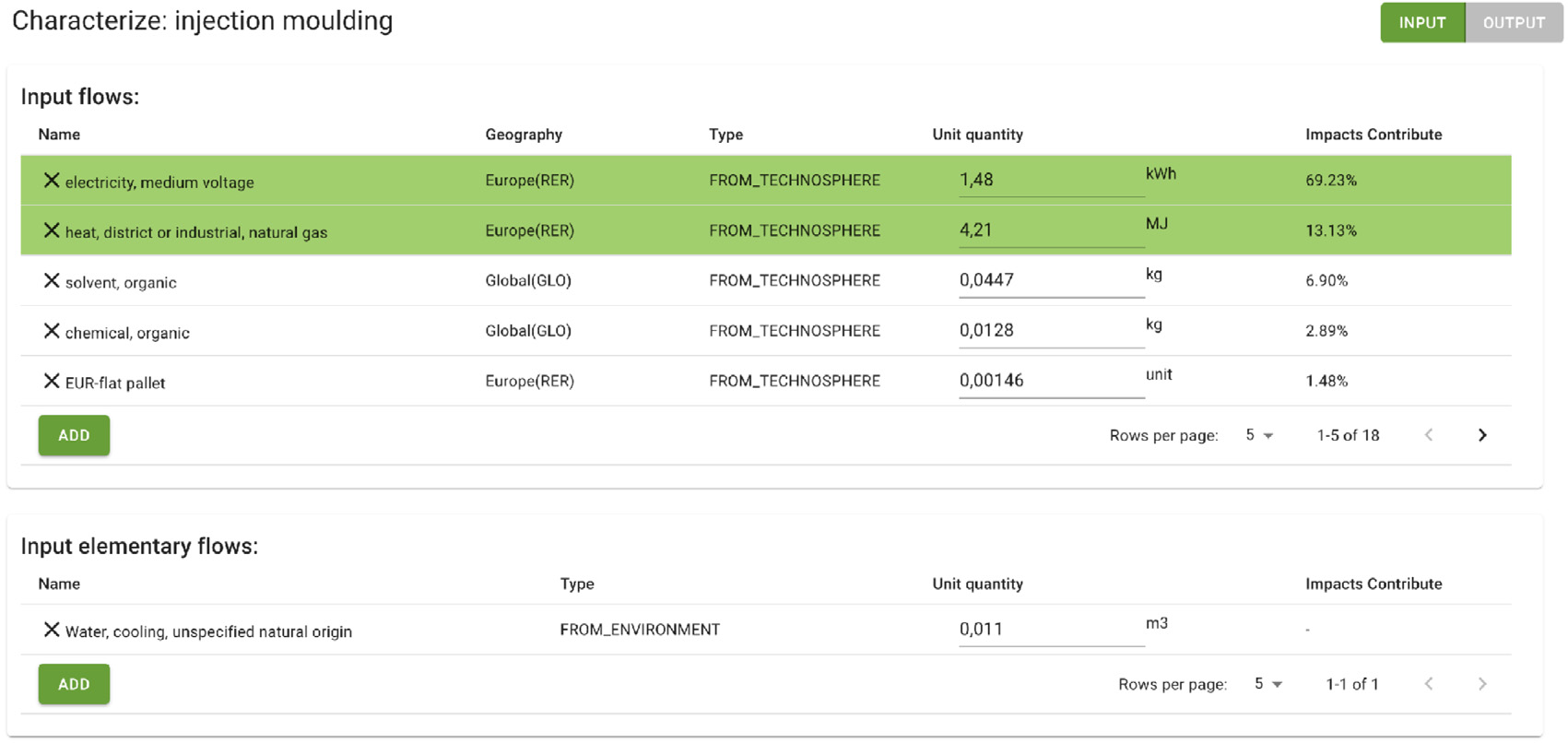
The Sustainability Assessment Application (SAA) is an evolution of an already existing application developed by SUPSI, i.e., Scuola Uni-versitaria Professionale della Svizzera Italiana, in previous EU projects (MANUTELLIGENCE [73] and MANUSQUARE [12]). SAA is an author-ing tool which allows users to characterize the processes provided by a production machine or a production line according to a LCA point of view. Fig. 2 shows an example of formalization of a set of processes. The tool enables the users to specify the SoS composition where the output flow (LCI output) of a certain process can be formalized as input flow (LCI input) for another process. The model underlying the SAA has been extensively described in Section 3, where the concept of PT emerges as an element related to a specific Functional Unit (i.e., the quantification of the system’s function analyzed by LCA), that is meant to quantify the function of the process under investigation (examples are: 1 h of milling process execution or 1 kg of removed steel for milling, 1 kg of injected plastic for injection molding, etc.; please refer to Fig. 2). Starting from the background data, retrieved from a LCIA

7

*V. Zambrano et al.*  *Array 14 (2022) 100176*



**Fig. 2.** List of the processes, for example belonging to a production line.



**Fig. 3.** Example of process characterization (LCI input and output) by the SSA. web version of this article.)

(For interpretation of the references to color in this figure legend, the reader is referred to the

|  |  |
| --- | --- |
| way of scaling up the production based on the expected sales. Right now, AgroIntelli assembles each robot on a made-to-order basis, at a single workstation, with predominantly manual work being required to assemble and interconnect all subcomponents and devices to each order. AgroIntelli would like to design and validate an automated man-ufacturing line for Robotti that can take the manufacturing blueprint and produce the robot.  Additionally, certain parts of Robotti, for example the air filter, the hydraulic oil, or the oil filter, need to be replaced during scheduled maintenance at different service intervals. When required, farmers also place orders for Robotti parts. Once the part orders from farmers are received, AgroIntelli needs to integrate the production of these parts into the manufacturing schedule. The updated production schedule will have to produce an expected delivery date for all the ordered parts. A DT of AgroIntelli’s manufacturing line helps the company per-forming an analysis of alternative scenarios for future factory con-struction. The projected manufacturing capacity, based on estimated sales and the manufacturing requirements of various parts based on sample Robotti usage scenarios, are key inputs of the factory DT. On the other hand, a feasible production schedule is the key output of the DT. The creation of the DT for the manufacturing line is done using the FMI [60] co-simulation standard. The key elements/workstations of the manufacturing line are modeled as FMUs. The DT for the manufacturing line is created by putting the FMUs together as one system and then co-simulated using Maestro [43]. The co-simulation *Algorithm* uses a configuration file for connecting the FMUs and performing a single co-simulation.  The factory DT is created from reusable FMUs which together form the *Model* of the DT. The Maestro co-simulation *Algorithm* becomes the *Algorithm* evaluating the factory DT *Model*. The evaluation of the factory *Model* by the Maestro *Algorithm* also requires *Data*. This *Data* come from the planned production schedules and the on-demand part orders from farmers.  In the context of agile manufacturing practices, it is important to keep the manufacturing line flexible and reroute the parts based on the | current demand. Therefore, FMUs need to be connected differently for co-simulating alternative manufacturing scenarios. One such scenario could be the creation of multiple models of Robotti. Another scenario could be the installation of parallel workstations to scale up the produc-tion of complicated machine parts. Each of these alternative scenarios can be explored by using one factory *Model* (implemented as multiple FMUs) and different co-simulation configurations. Therefore, the Mae-stro co-simulation *Algorithm* is reusable for all co-simulation models. Additionally, different manufacturers can bring in their co-simulation models and perform co-simulation using Maestro co-simulation *Algo-rithm*. The factory *Model* can also be reused by AgroIntelli to perform different design scenarios for the planned manufacturing line. In other words, the same *Model* can be used to perform Design Space Exploration (DSE) for designing an optimum manufacturing line [15]. Since each FMU represents a factory workstation, there is a potential for reusing single FMUs to create new co-simulation scenarios of other factory configurations/factories. As a result, a DT created for AgroIntelli is reusable within the broader manufacturing industry too. This has been made possible by the clean separation of *Data*, *Algorithm*, and *Model* of the factory DT using the DIGITbrain approach.  *6.2. Reduced order models for injection molding*  A physically based DT for a thermoplastic injection molding consists of the digitalization of a thermoplastic material injection molding process for the quick initial set-up of process parameters and the opti-mization during production. The DT is based on the offline exploitation of state-of-the-art numerical simulation tools to model the relevant fluid-dynamics, heat transfer and thermo-mechanical physical mecha-nisms, as well as subsequent encapsulation of relevant Key Performance Indicators (KPIs) into ROMs that are made available for the user to get a real time virtual try-out and estimated production quality monitoring. Within the DIGITbrain experiment performed for Inymon, a Spanish injection molding company, current state-of-the-art simulation models |

8

*V. Zambrano et al.*  *Array 14 (2022) 100176*

representing the physics of the material transformation process, in-cluding transient flow fluid-dynamics, heat transfer by conduction and convention, and solid thermo-mechanics were used. Specific material relations, such as non-Newtonian rheology and equations of state, reproduce the material’s behavior. These models are expressed as a set of differential equations corresponding to the conservative laws (i.e., mass, moment, energy) in fluid and solid domains. These phys-ical equations are solved using Volume of Fluid (VoF)/FEM by the Moldex3D commercial program [75]. Furthermore, an offline execution of a simulation DoE is carried out to allow the virtual exploration of the *Model* to generate a database of simulation results, covering suitable variation ranges of the processing parameters (i.e., melt temperature, filling time/ram speed, switch over point, packing pressure and time, coolant temperature and flow rate, cooling time, etc.).

As mentioned previously, online process modeling for quality con-trol requires real-time simulation capabilities, which cannot be achieved by executing the computationally intensive physically based simulation models (as these require hours to run) within a real cycle time (i.e., *<*1 min). Creating ROMs has proved to be a valid approach to provide real-time responses to Physically Based Simulations (PBS). A ROM retains only the relevant information required for quality evaluation (i.e., selected KPIs) and process control. Among the different techniques available for ROM generation, *a posteriori* or non-intrusive techniques have a higher potential, especially combined with real *Data*. CAELIA™’s ROM generation, based on the factorization of the information through TRD approach, results into a sequence of products of separable 1D functions that can be solved *on the fly* (see Eq. (1)), allowing having a transfer function directly related to input parameters, independently [56].

Within the scope of the DIGITbrain Project, the ROM generation solution, used to build the ROMs, has been deployed in the cloud-based platform as a job. Moreover, the CAELIA™ App including a Graphical User Interface (GUI) has also been deployed on a Virtual Machine (VM) instance on the Platform. The GUI allows the user to compose tuples of ROMs (as DIGITbrain *Models*), input *Data*, as well as evaluation and optimization *Algorithms* to try finding the parameters describing the best injection molding process.

ROMs are specific of each injection process (IP instance), as they are built on physically based simulation *Data* generated for a specific physical instance, which makes the reusability of ROMs anything but straightforward. Nevertheless, the algorithms used to build ROMs and the CAELIA™ App are generic, so that different users can employ them and upload their ROMs and *Data* from other injection molding lines. A possible way to reuse ROMs would be the development of an expert system that, by analyzing a database of ROMs, generated for different injection molding processes, could extract rules of dependencies for typical results obtained from defined process conditions (knowledge-based reasoning). As a further step, the aforementioned expert system could also, based on the characteristics of the part to be injected, be able to create a new ROM backbone by proposing 1D functions for the TRD factorization, that represent the dependency of a result on a specific process condition. This *Model* could be used as a low fidelity *Model* to analyze tendencies or as the basis for a new adjustment, based on PBS or measured *Data*.

*6.3. System of systems models for laser-cutting and forming of aluminum*

Creation of a simulation tool which allows the end user to create alternative layouts of production lines for forming aluminum sheets and compare the performances of the different solutions has high importance. The main limitations that reflect the current status of most EU manufacturing SMEs in the utilization of DTs of production lines, are the following:

• the inability to perform real-time *what-if* analysis based on the actual plant *Data*, that strongly impairs the ability to promptly tackle production needs and constraints,

9

*V. Zambrano et al.*  *Array 14 (2022) 100176*

increased production and the high standards of consumers’ expectations in the Industry 4.0 (I4.0) era encourage the introduction of Digital Twins (DTs) in the manufacturing industry, especially to help SMEs forecasting and optimizing their systems (e.g., processes, machinery, production lines) under different operating conditions. Offering MaaS on an integrated digital platform is an efficient way to achieve that. In the paper, it was highlighted how DIGITbrain’s overall concept supports the reusability of *Models* by separating the fundamental build-ing blocks of DTs, including the *Model* itself, its *Algorithms* and *Data*, with a rich set of metadata. Moreover, given the definition of a DT as a virtual model of a physical system, models represent a crucial asset for industrial digitalization and the introduction of DTs in manufacturing processes. We further detailed some model types that are currently supported by the DIGITbrain Platform, such as co-simulation, Artificial Intelligence (AI) and System of Systems (SoS) models. Moreover, we detailed three representative case studies, where practical applications of DTs were delivered to manufacturing SMEs and their end-users. Further work is being conducted towards the generation of im-proved models for several applications. The models will be published on the project’s integrated platform which is currently being upgraded to grant scalability, parallelization and efficiency.

**Declaration of competing interest**

The authors declare that they have no known competing finan-cial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Acknowledgment**

This work was funded by the DIGITbrain – Digital twins bring-ing agility and innovation to manufacturing SMEs, by empowering a network of DIHs with an integrated digital platform that enables Manufacturing as a Service project, No. 952071, European Commission (EU H2020).

**References**

[1] [Grieves M. Digital twin: manufacturing excellence through virtual factory](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb1)  [replication. White paper 1, Florida Institute of Technology; 2014, p. 1–7.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb1)

[2] [European Commission. Digital transformation scoreboard 2018. EU busines](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb1)[ses go digital: Opportunities, outcomes and uptake. Luxembourg: Publications Office of the European Union; 2018, p. 10–2826, 10 (691861).](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb2)

[3] [Semeraro C, Lezoche M, Panetto H, Dassisti M. Digital twin paradigm: A](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb3)  [systematic literature review. Comput Ind 2021;130:103469.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb3)

[4] [Jones D, Snider C, Nassehi A, Yon J, Hicks B. Characterisin](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb3)[g the digital twin: A](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb4)  [systematic literature review. CIRP J Manuf Sci Technol 2020;29:36–52.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb4)

[5] [Fisher O, Watson N, Porcu L, Bacon D, Rigley M, Gomes RL. Cloud manuf](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb4)[acturing as a sustainable process manufacturing route. J Manuf Syst 2018;47:53–68.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb5)

[6] [Kusiak A. Service manufacturing: Basic concepts and technologies. J Manuf](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb5) [Syst](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb6)  [2019;52:198–204.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb6)

[7] [Kusiak](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb6)  [A.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb6)  [Servi](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb6)[ce](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb7)  [manufacturing](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb7)  [=](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb7)  [process-as-a-service](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb7)  [+](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb7)  [manufacturing](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb7)  [operations-as-a-service. J Intell Manuf 2020;31(1):1–2.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb7)

[8] [Rauschecker U, Meier M, Muckenhirn R, Yip ALK, J](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb7)[agadeesan AP, Corney J.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb8)

[Cloud-based manufacturing-as-a-service environment for customized products.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb8)

[In: EChallenges E-2011 conference proceedings. IIMC International Information Management Corporation; 2011.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb8)

[9] [Meier M, Seidelmann J, Mezgár I](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb8)[. ManuCloud: the next-generation manufacturing](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb9)  [as a service environment. ERCIM News 2010;33–4.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb9)

[10] [Hasan M, Starly B. Decentralized cloud manufa](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb9)[cturing-as-a-service (CMaaS) platform architecture with configurable digital assets. J Manuf Syst 2020;56:157–74.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb10)

[11] [Kiss T. A cloud/HPC platform and marketplace for manufacturing SMEs. In: 11th](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb11)  [international workshop on science gateways, IWSG 2019. 2019.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb11)

[12] [Landolfi G, Barni A, Izzo G, Fontana A, Bettoni A. A MaaS pl](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb11)[atform architec-ture supporting data sovereignty in sustainability assessment of manufacturing systems. Procedia Manuf 2019;38:548–55.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb12)

[13] [Kiss T, Kacsuk P, Kovács J, Rakoczi](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb12) [B, Hajnal A, Farkas A, Gesmier G, Terstyanszky G. Micado—microservice-based cloud application-level dynamic orchestrator. Future Gener Comput Syst 2019;94:937–46.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb13)

[14] [Cloud orchestration at the level of application iCal. About](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb13) - COLA project. 2017, <https://project-cola.eu/cola-project/>.

10

*V. Zambrano et al.*  *Array 14 (2022) 100176*

[42] [Blochwitz T, Otter M, Akesson J, Arnold M, Clauss C, Elmqvist H, Friedrich M,](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb42)  [Junghanns A, Mauss J, Neumerkel D, Olsson H, Viel A. The functional mockup in-](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb42) [terface 2.0: The standard for tool independent exchange of simulation models. In:](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb42)  [Proceedings of the 9th international modelica conference. Munich, Germany: The](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb42)  [Modelica Association, Linköping University Electronic Press; 2012, p. 173–84.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb42) [43] [Thule C, Lausdahl K, Gomes C, Meisl G, Larsen PG. Maestro: The INTO-CPS c](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb42)o- simulation framework. Simul Model Pract Theory 2019;92:45–61. [http://dx.doi.](http://dx.doi.org/10.1016/j.simpat.2018.12.005)

[org/10.1016/j.simpat.2018.12.005](http://dx.doi.org/10.1016/j.simpat.2018.12.005), URL [http://www.sciencedirect.com/science/](http://www.sciencedirect.com/science/article/pii/S1569190X1830193X) [article/pii/S1569190X1830193X.](http://dx.doi.org/10.1016/j.simpat.2018.12.005)

[44] [Popper J, Hermann J, Cui K, B](http://www.sciencedirect.com/science/article/pii/S1569190X1830193X)[ergweiler S, Weyer S, Ruskowski M, Wang M, Guang L, Hu Y, Kueh V, Wang D, Madigan M, Oppermann I, Ryu S-M, Kim L, Wei S, Li R, Du X, Shang Y, Thonet G. Artificial intelligence across industries. Tech. rep., IEC Whitepaper; 2018, p. 93.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb44)

[45] [Rojas R. The backpropagation algorithm](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb44)[. In: Neural networks. Springer; 1996,](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb45)  [p. 149–82.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb45)

[46] [Bottou L.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb45) [Stochastic gradient descent tricks. In: Neural networks: Tricks of the](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb46)  [trade. Springer; 2012, p. 421–36.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb46)

[47] [Rowley CW, Colonius T, Murray](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb46) [RM. Model reduction for compressible flows](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [using POD and Galerkin projection. Physica D 2004;189(1–2):115–29.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)

[48] [Rozza](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [G,](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [Veroy](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [K.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [On](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [the](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [stability](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [of](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [the](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [reduced](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [basis](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)  [me](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb47)[thod](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb48)  [for](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb48)  [Stokes equations in parametrized domains. Comput Methods Appl Mech Engrg](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb48)  [2007;196(7):1244–60.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb48)

[49] [Baur U,](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb48)  [Benner P,](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb48)  [Feng](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb49)  [L. Model](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb49)  [order reduction](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb49)  [for linear and](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb49)  [non-](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb49) [linear systems: a system-theoretic perspective. Arch Comput Methods Eng](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb49)  [2014;21(4):331–58.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb49)

[50] [Chinesta F, Cueto E](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb49)[. PGD-based modeling of materials, structures and processes.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb50)  [Springer; 2014.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb50)

[51] [El Halabi F, Go](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb50)[nzalez D, Chico-Roca A, Doblare M. Multiparametric response sur-face construction by means of proper generalized decomposition: An extension of the PARAFAC procedure. Comput Methods Appl Mech Engrg 2013;253:543–57.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb51) [52] [Cichocki A, Zdunek R, Phan AH, Amari S-i. Nonnegative matrix and tensor](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb51) [factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons; 2009.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb52)

[53] [Willcox K, Peraire J. Balanced model reduction via the proper orthogonal](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb53)  [decomposition. AIAA J 2002;40(11):2323–30.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb53)

[54] [Ibáñez R, Abisset-Chavanne E, Ammar A,](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb53) [González D, Cueto E, Huerta A,](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [Duval](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [JL,](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [Chinesta](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [F.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [A](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [multidimensional](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [data-driven](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [sparse](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [identifica-](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54) [technique:](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [The](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [sparse](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [proper](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [generalized](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [decomposition.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)  [Complexity tion](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)   
 [2018;2018:1–11.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)

[55] [Izquierdo S, Ro](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb54)dr’iguez R, Zambrano V. TWINKLE: A digital-twin-building kernel for real-time computer-aided engineering. 2020, [https://github.com/ caeliaITAINNOVA/Twinkle](https://github.com/caeliaITAINNOVA/Twinkle).

[56] [Zambrano V, Rodríguez-B](https://github.com/caeliaITAINNOVA/Twinkle)arrachina R, Calvo S, Izquierdo S. TWINKLE: A digital-twin-building kernel for real-time computer-aided engineering. SoftwareX 2020;11:100419. <http://dx.doi.org/10.1016/j.softx.2020.100419>, URL [https:// www.sciencedirect.com/science/article/pii/S2352711019300664](https://www.sciencedirect.com/science/article/pii/S2352711019300664)[.](http://dx.doi.org/10.1016/j.softx.2020.100419)

[57] [Ayodele T. Types of machine learning algorithms. In: New adva](https://www.sciencedirect.com/science/article/pii/S2352711019300664)nces in machine learning. InTech; 2010, p. 19–48. <http://dx.doi.org/10.5772/9385>, Ch. 3.

[58] International Organization for Sta[ndardization. Environmental m](http://dx.doi.org/10.5772/9385)anagement –Life Cycle Assessment – Principles and framework. URL [https://www.iso.org/ standard/37456.html](https://www.iso.org/standard/37456.html).

[59] [McClure C. Softwar](https://www.iso.org/standard/37456.html)[e reuse - a standards-based guide. Wiley-IEEE Computer](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb59)  [Society Press; 2001.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb59)

[60] [Functional mock-up](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb59) interface. <https://fmi-standard.org/tools/>.

[61] [Bagnato A, Brosse E, Quadri](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb61) [I, Sadovykh A. INTO-CPS: A](https://fmi-standard.org/tools/)[n integrated ‘‘tool chain’’ for comprehensive: model-based design of cyber-physical systems. In: ICSSEA 2015 proceedings. 2015, p. 31–5, This publication is part of the Horizon 2020 project: Integrated Tool chain for model-based design of CPSs (INTO-CPS), project/GA number 644047.; null ; Conference date: 27-05-2015 Through 29-05-2015.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb61)

[62] Modelio: Open source development environment. 2021, [https://www.modelio.](https://www.modelio.org/)  [org/](https://www.modelio.org/).

[63] [Lars](https://www.modelio.org/)[en PG, Battle N, Ferreira M, Fitzgerald J, Lausdahl K, Verhoef M. The overture initiative integrating tools for VDM. ACM SIGSOFT Softw Eng Notes 2010;35(1):1–6.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb63)

[64] [Fitzgerald J, Ga](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb63)mble C, Larsen P, Pierce K, Woodcock J. Cyber-physical systems design: Formal foundations, methods and integrated tool chains. In: Proceedings of the 2015 IEEE/ACM 3rd FME workshop on formal methods in software engi-neering (FormaliSE). IEEE; 2015, p. 40–6. [http://dx.doi.org/10.1109/FormaliSE.](http://dx.doi.org/10.1109/FormaliSE.2015.14)

[2015.14](http://dx.doi.org/10.1109/FormaliSE.2015.14), This publication is part of the H[orizon 2020 project: Integrated Tool chain fo](http://dx.doi.org/10.1109/FormaliSE.2015.14)r model-based design of CPSs (INTO-CPS), project/GA number 644047.; null ; Conference date: 18-05-2015 Through 18-05-2015.

[65] [Larsen PG, Lausdahl K, Battle N, Fitzgerald J, Wolff S, Sahara S, Verhoef M, Tran-Jørgensen PW, Oda T, Chisholm P. VDM-10 language manual. 2010.](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb65)

[66] [Fritzson P, Pop A, Abdelhak K, Ashgar A, Bachmann B, Braun W, Bousk](http://refhub.elsevier.com/S2590-0056(22)00035-2/sb65)ela D, Braun R, Buffoni L, Casella F, Castro R, Franke R, Fritzson D, Gebremed-hin M, Heuermann A, Lie B, Mengist A, Mikelsons L, Moudgalya K, Ochel L, Palanisamy A, Ruge V, Schamai W, Sjölund M, Thiele B, Tinnerholm J, Östlund P.

The OpenModelica integrated environment for modeling, simulation, and model-based development. Model Identif Control 2020;41(4):241–95. [http://dx.doi.org/ 10.4173/mic.2020.4.1](http://dx.doi.org/10.4173/mic.2020.4.1).

11

*V. Zambrano et al.*  *Array 14 (2022) 100176*

**Giuseppe Landolfi** works at the Institute of Systems and Technologies for Sustainable Production of the Department of Innovative Technologies within the Sustainable Production Systems Laboratory (SPS). He has been involved in several National and European projects as Software Engineer with a supervisory role of a group of developers. He is lecturer in several courses such as Introduction of Programming, Introduction of Object Programming and Industry 4.0.

**Alessandro Fontana** M. Sc. Eng. Alessandro Fontana is Lecturer & Researcher at SUPSI. Graduated in Material Engineering, he had a six-year industrial experience in Life Cycle Assessment of products and manufacturing processes within Legrand Group. Member of Sustainable Production Systems Lab of SUPSI, his main research domain is sustainability assessment in manufacturing. He teaches Industrial Sustainability at Master and Bachelor level.

**Manuel Laspalas** Ph.D. in Industrial Engineering. He has been leading ITAINNOVA’s research lines on ‘‘Multiscale Analysis and Simulation of Unconventional Transformation Processes’’ (2013–2015) and ‘‘R+D+i in Polymers’’ (2017–2020). He currently performs functions of Project Manager in strategic projects for ITAINNOVA and leads the research line on ‘‘Sustainable Material Transformation Processes’’ (2021–2024).

**Jibinraj Antony** completed his M.Sc. in Mechatronics from University of Siegen, Germany, with focus on Deep Learning. Presently he is working as researcher at DFKI focusing on the digital transformation of SMEs. He is also part of the Mittelstand Digital Zentrum (MDZ), offering consultation for SMEs in their digitalization and AI implementation efforts.

**Valerie Poser** is a researcher at DFKI, focusing on software engineering in cloud environments. She received her B.Sc. and M.Sc. in Computer Science from Saarland University in the field of classical AI planning. Her current research activities address machine learning in cloud environments, with the goal to facilitate the application of state-of-the-art ML tools for industrial applications.

**Tamas Kiss** is a Professor of Distributed Computing at the School of Computer Science and Engineering and Director of the Centre for Parallel Computing at the University of Westminster. He is also Editor in Chief of the Journal of Grid Computing, published by Springer Nature. He attracted over £50 Million research funding and led several national and European research projects.

**Simon Bergweiler** received a diploma in the field of applied computer science at Trier University of Applied Sciences. Since then, he has been working on national and international projects at the German Research Center for Artificial Intelligence (DFKI). The focus of his current research activities as a senior engineer is on the Cognitive Factory and Human-Centered Subsidiary Industrial Production. In the context of this paper, Machine Learning models as well as the use of methods of AI were considered in terms of Manufacturing as a Service (MaaS).

12