



HAL
open science

Data-driven invariant modelling patterns for digital twin design

Concetta Semeraro, Mario Lezoche, Hervé Panetto, Michele Dassisti

► To cite this version:

Concetta Semeraro, Mario Lezoche, Hervé Panetto, Michele Dassisti. Data-driven invariant modelling patterns for digital twin design. *Journal of Industrial Information Integration*, 2023, 31, pp.100424. 10.1016/j.jii.2022.100424 . hal-03888725

HAL Id: hal-03888725

<https://hal.science/hal-03888725>

Submitted on 7 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Data-driven Invariant Modelling Patterns for Digital Twin Design

Concetta Semeraro^{1,2,3*}, Mario Lezoche², Hervé Panetto²,
and Michele Dassisti³

¹ *Department of Industrial and Management Engineering, University of Sharjah, Sharjah, United Arab Emirates (e-mail: csemeraro@sharjah.ac.ae)*

² *University of Lorraine, CNRS, CRAN, France (e-mail: {mario.lezoche, herve.panetto}@univ-lorraine.fr)*

³ *Department of Mechanics, Management & Mathematics (DMMM), Polytechnic University of Bari, Bari, Italy (e-mail: michele.dassisti@poliba.it)*

Abstract

The Digital Twin (DT) is one of the most promising technologies in the digital transformation market. A digital twin is a virtual copy of a physical system that emulates its behaviour to predict failures and opportunities for change, prescribe actions in real-time, and optimise and/or mitigate unexpected events. Modelling the virtual copy of a physical system is a rather complex task and requires the availability of a large amount of information and a set of accurate models that adequately represent the reality to model. At present, the modelling depends on the specific use case. Hence, the need to design a modelling solution suitable for virtual reality modelling in the context of a digital twin. The paper proposes a new approach to design a DT by endeavouring the concept of "modelling patterns" and their invariance property. Modelling patterns are here thought of as data-driven, as they can be derived autonomously from data using a specific approach devised to reach an invariance feature, to allow these to be used (and re-used) in modelling situations and/or problems with any given degree of similarity.

The potentialities of invariance modelling patterns are proved here by the grace of a real industrial application, where a dedicated DT has been built using the approach here proposed.

Keywords: Invariance; Modelling patterns; Digital Twin; Data-driven; Cyber-physical Systems; Die-casting;

ABBREVIATIONS

| | |
|--------------|---|
| C_1 | Course of the plunger in the first phase |
| C_2 | Course of the plunger in the second phase |
| CC | Multiplied course |
| CTS | Critical-to-solution variables |
| DT | Digital Twin |
| $c_{l,al,i}$ | Latent heat of aluminium |
| $c_{p,al,i}$ | Specific heat of aluminium |
| FCA | Formal concept analysis |
| $F_{inj,i}$ | Injection force |
| FC | Clamping force |
| h_0 | Specific enthalpy of environment |
| h_{al} | Specific enthalpy of aluminium |
| MES | Manufacturing Execution Systems |
| P_0 | Environmental pressure |
| P_{a1} | Accumulator pressure in the first stage |
| P_{a2} | Accumulator pressure in the second stage |
| P_{aM} | Accumulator pressure in the multiplied stage |
| PF | Final pressure |
| PLC | Programmable logical Controller |
| PM | Multiplied pressure |
| PS | Specific pressure |
| $Q_{c,i}$ | Energy for casting |
| $Q_{f,i}$ | Energy for melting |
| $Q_{l,i}$ | Energy for the transformation |
| RCA | Relation concept analysis |
| s_0 | Specific entropy of the environment |
| s_{al} | Specific entropy of aluminium |
| S_{inj} | Injection surface |
| SM | Thickness of the product |
| T_0 | Environmental temperature |
| T_1 | Time of injection in the first phase |
| T_2 | Time of injection in the first phase |
| T_f | Melting temperature |
| T_{mo} | Mould temperature |
| T_s | Temperature of the ingots at the entry of the furnace |
| $T_{al,i}$ | Injection temperature of aluminium |
| T_l | Temperature loss during the injection stage |
| TC | Cycle time |
| V_1 | Speed of the plunger in the first phase |
| V_2 | Speed of the plunger in the second phase |
| $V_{al,i}$ | Volume of Aluminium |
| V_l | Volume of Aluminium Loss during the injection stage |
| ρ_{al} | Density of Aluminium |

INTRODUCTION

The product and production system are becoming increasingly complex as the number of components, the frequency of market demand changes, and the need for related innovation increases. Digital representations are a significant opportunity to manage this complexity and improve decision-making accuracy using existing simulation and emulation tools [1]. A digital representation -above all, the digital twin - bridges the gap between the physical and virtual system, improving the interpretation of reality using sound data collection and interpretation [2]. A digital twin is *“a set of adaptive models that emulate the behaviour of a physical system in a virtual system getting real-time data to update itself along its life cycle. The digital twin replicates the physical system to predict failures and opportunities for changing, to prescribe real-time actions for optimising and/or mitigating unexpected events observing and evaluating the operating profile system”* [3]. It can be a model of a component, a system of components, or systems, such as pumps, motors, power plants, or production lines, etc. It requires a set of models appropriately representing the physical resources and the processes knowledge to usefully perform decisional support. Modelling such a digital copy of the physical system to perform real-time decision validation and process optimisation is quite a complex task, due to the number of variables and their relationships, also taking into account that functional relationships and constitutive laws are barely available on real components. Hopefully, a large amount of information may help to pursue the DT modelling goal, but better it may serve to recognise appropriate modelling patterns that synthesise information and extract the operational semantics of the modelled elements [4], [5]. Typically, it is hard to construct accurate digital models using traditional model-based approaches because of the recalled complexity and lack of appropriate knowledge [6]. The disunity in the literature on modelling the physical systems leads to the need to devise a new modelling solution to apply in DTs design applications to save time and effort and improve the efficacy [7].

This paper proposes a novel methodology of a more structured and reliable modelling approach for DTs by endeavouring the invariance concept. The idea is to detect and formalise automatically invariant modelling patterns to interpret reality from historical data (of any granularity and type). Hereof, the research question that we raise and discuss is: *“How can the invariance concept may improve the modelling effort to implement DT?” - as a consequence of this ‘How to achieve invariant modelling patterns?’*. The rationale behind this is that by appropriately building and recognising invariant modelling patterns, it is possible to infer knowledge of a system’s behaviour from reality (described, as an example, by data) and thus to describe (or even emulate) a system independently of its application context, thus capturing the essence of its behaviour and thus maximising the efficiency of the digital image of reality. Predefined data-driven patterns may represent invaluable knowledge from extracting digital models for different applications. Invariance in our idea guarantees the model's significativity by capturing the essence of any system’s behaviour independently of the specific application domain. Once this statement is proven, the question becomes how to derive criteria and a methodology to self-detect invariant modelling patterns-data-driven that can be used (and re-used) to create digital models of different systems or processes [8].

The paper presents in section 1 an overview of the existing literature approaches to model a digital twin independently of the specific context of their use and drawbacks. After performing an exhaustive literature review, we discuss in section 2 the research question put into conceiving and structuring the modelling approach - in the form of modelling patterns - to allow more straightforward design,

modelling, development, and maintenance of digital twins. Section 3 formalises the approach to detect and formalise data-driven invariant modelling patterns. Finally, mainly as proof of the approach presented, a case study is presented in section 4, where all possible data-driven patterns define how these can be used along with different applications. Section 5 presents the conclusions.

1. DIGITAL TWIN MODELLING APPROACHES

Modelling manufacturing systems to design their Digital Twins requires efficient approaches to grab the variables' complexity, their interrelationships and the related links to performances. Numerous methods and tools have been proposed and developed in the literature to achieve this aim.

A model is a representation of a system for a specific purpose [9] developed to understand and formalise a system [10], i.e. a collection of entities (e.g., people or machines) that act and interact together toward the accomplishment of some logical end [11]. Existing modelling approaches to design digital twins can be distinguished into three broad macro-classes: model-based, data-driven and hybrid approaches (that involve both models and data) [12], [13] as summarised in Figure 1 and explained below.

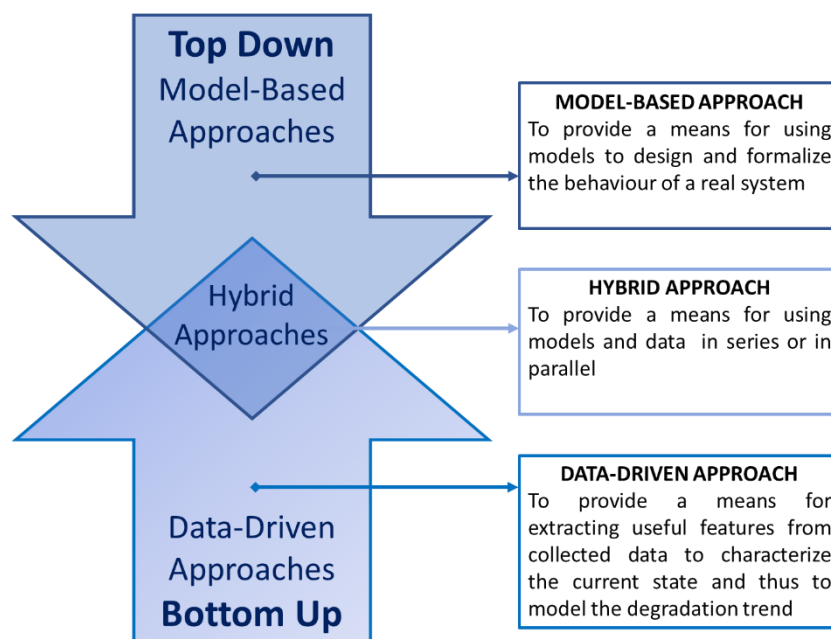


Figure 1: Main classes of DT modelling approaches.

Model-based approaches use mathematical equations to incorporate physical knowledge about a system's degradation/failure behaviours whenever available [14]. A model-based approach is considered a top-down approach. Top-down approaches interpret and incorporate incoming information based on prior knowledge, physical understanding and experiences into a set of models. [15]. Model-based approaches need small data but require mathematical models [2]. Model-based systems engineering (MBSE) and Model-driven architecture (MDA) are the main model-based approaches in digital twin applications. Model-based systems engineering (MBSE) is defined as: *“formalised application of modelling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life-cycle phases”* [16]. Aiming to support MBSE, diverse design languages

and tools imposed into the whole design cycle are devised to precisely specify the design specification, including requirements, function and behaviour, such as System-modelling language (SysML) [17], Rational Rhapsody Developer, Modelica [18]. The Digital Twin is becoming increasingly relevant to Model-based systems engineering (MBSE) [19]. It extends the MBSE from engineering and manufacturing to the operation and service phases [20]. [21] presents the digital twin concept within the model-based systems engineering (MBSE) context. The context of the operation of the digital twin involves an instrumented testbed in which model-based systems engineering (MBSE) tools (e.g., system modelling and verification tools) and operational scenario simulations (e.g., discrete event simulations, agent-based simulations) are used to explore the behaviour of virtual prototypes in a what-if simulation [21], [22]. The model-driven architecture (MDA) is a software-designed approach developed by the Object Management Group (OMG) [23]. MDA provides guidelines for structuring software specifications that are expressed as models, providing three types of models from three different viewpoints: the independent computation model (CIM), the platform-independent model (PIM), and the platform-specific model (PSM). The key idea of MDA is the separation between upper-level business and the underlying implementation. In recent years, an increasing trend has been applying MDA to DTs' design. Typical works using MDA for digital twins are discussed in [24], [25]. Model-based approaches rely on the use of models to simulate the system's behaviour in different operating conditions, but these models are not easy to develop and keep updated during the system life-cycle [26]. Therefore, detailed domain knowledge of underlying the degradation processes leading to failure is required to develop physics-based models. The model-based approaches require a priori knowledge of processes, which are usually developed based on fundamental comprehension of the physics of the process to be monitored [10]. The model is developed based on the understanding and the formalisation of the physics of the process. The success of model-based approaches depends on the model's accuracy, fidelity, and robustness [27]. These characteristics can be accurate when the degradation physics knowledge is sufficient and relevant. Therefore, the degradation process of a system cannot be described directly in a precise way due to the complexity and the variety that can characterise a particular system [17].

In contrast to the model-based approaches, where a-priori knowledge about the process is needed, in data-driven approaches, a large amount of historical data [27] is required to discover tacit and explicit knowledge from data collection. Data-driven approaches are considered bottom-up because the models are built based on retrieving sensory information from the external environment [28], [29]. Data-driven approaches are designed to detect hidden patterns and knowledge by analysing a massive amount of historical data [30]. The data-driven approaches can be classified into supervised and unsupervised [31]. Supervised is the task of inferring a function from labelled training data. It aims to build a model that can correctly predict the output of an unseen instance by observing a set of labelled instances [32]. Supervised data mining attempts to explain the behaviour of the target as a function of a set of independent attributes [33]. The principal supervised mining techniques are classification and regression [34]. These techniques predict target categories [33] and numeric values [34]. The primary strength of the supervised approaches is that these approaches do not require a high level of domain knowledge [35]. The supervised learning across the digital twin is applied for the system's failure prediction (Asimov et al., 2018) or the prediction of the remaining useful life (RUL) of the physical twin (Z. Liu et al., 2018). The major weaknesses are the massive amounts of data needed for training a reliable model [36]. Instead, unsupervised learning [37] only discovers an internal representation from input data. The principal unsupervised mining techniques are clustering, association rule mining (for example, Formal concept analysis (FCA) technique), and sequential

pattern discovery. The clustering technique aims to split a set of data into subsets (clusters) with similar characteristics [37]. The Association technique aims to detect the probability of the co-occurrence of items in a collection [38]. The sequential patterns discovery technique aims to identify associations or patterns over time [39]. The goal is to model the states of a process generating the sequence trends and the report deviation over time. In digital twins, these techniques are used for creating autonomous clusters for different working regimes to analyse machine conditions (Lee et al., 2014a), (Banerjee et al., 2017), (Ding et al., 2019). The primary strength of the unsupervised data-driven approaches is the possibility of discovering tacit knowledge from a set of data, such as unknown faults and operation patterns [40]. The major weaknesses are the massive amounts of data for training a reliable model, and the results and the performances depend on the quality of training data [41]. Unsupervised data-driven approaches require a-priori knowledge model [36] to find valuable knowledge. Data-driven approaches allow integrating parameters across different domains (e.g., product, process, and logistics) into models that would be difficult to build with the traditional model-based approaches. Data-driven approaches extract helpful features from collected data to characterise the current state and, thus, the degradation trend. Using data-driven approaches is possible to transform data into relevant information and reliable behavioural models [42], [43]. Data-driven approaches involve massive data sets and high dimensionality. The major limitation of data-driven approaches is the prerequisite of sufficient training data relevant to the failure/degradation under study.

Recently, there has been a growing interest in data-driven and model-based integration approaches to provide accurate results. The combination of data collected during the system with the system's physical models can address the lack of model-based and data-driven approaches [44], [45], [46]. The existing **hybrid approaches** can mainly be categorised into series and parallel approaches. A series approach combines a physics-based model with prior knowledge about the process and a data-driven model that serves as a state estimator of unmeasured process parameters that are hard to model by first principles [47]. In [48], a model-based approach is used to define the failure threshold, while a data-driven approach calibrates the model to make the results more accurate and narrow the uncertainty of model results [49]. A parallel approach can benefit from the advantages of physics-based and data-driven models, such that the output of the resulting hybrid model is more accurate [50], [47]. In [51], a data-driven approach is applied to infer a measurement model, while a model-based approach describes the system's behaviour. Hybrid approaches combine the advantages of both approaches [41], [52], [53]. The existing hybrid approaches are series and parallel approaches. A hybrid series approach obtains the prior knowledge of the physics-based model in the first step and attains unmeasured process parameters by data-driven approaches in the second step. A hybrid parallel approach learns from model-based and data-driven approaches at the same time. Despite the advantages and the potentials of the hybrid approaches, the main obstacle is the lack of a standard framework or methodology to apply [41] and the complexity of the selection, combination and parameter tuning of various methods. The advantages and disadvantages of each approach are summed up in Table 1.

Table 1: Comparison of Modelling Approaches for DT design.

| DT Macro-Approaches | PRO'S | CON'S |
|---------------------|---|---|
| Model-based | <ul style="list-style-type: none"> • Accurate to describe degradation/failure behaviours [54]. • Provide knowledge about the degradation/ failure [55]. • Require a small amount of data [55]. • Integrate the physical understanding of the system into the fault detection and diagnosis process [56]. • Incorporate the interactions with the environment into the fault detection and diagnosis process [54]. | <ul style="list-style-type: none"> • Hard to observe the degradation process directly [56]. • Require an explicit mathematical model and rely on a set of assumptions and approximations [55]. • The performances depend on the robustness of the model [56]. • Physics-based models developed in laboratories need to be adapted to simulate the real environment in practice [55]. • The estimation becomes more difficult if the system complexity increases [54]. • Models rely on a set of assumptions and approximations [56]. • The model-based approaches are not suitable for large-scale and complex systems [55]. |
| Data-driven | <ul style="list-style-type: none"> • Un-necessary to model various degradation/failure physics precisely [57]. • Require little domain knowledge [41]. • Easier to implement [41]. • Handle high-dimensional and correlated process variables [57]. • Suitable for complex systems and large-scale systems [58]. • Automatic discovery of patterns or models from a data set [57]. • Discovery of tacit knowledge in the data [57]. • They can reduce time and cost since they do not require the development of models [58]. | <ul style="list-style-type: none"> • Reliance on relevant and quality data [58]. • The quality of the results and the performances rely on training data and may degrade once the system is working in an unknown condition (i.e. outside of the training data) or is affected by unknown faults [41]. • High effort in preprocessing step to extract useful information from data with a high computational cost [41]. |
| Hybrid | <ul style="list-style-type: none"> • Combine the advantages of model-based and data-driven approaches to provide better diagnostic systems [41]. | <ul style="list-style-type: none"> • The complexity of the selection, combination and parameter tuning of various methods [41]. |

2. INVARIANT MODELLING PATTERNS

A summary of approaches for modelling digital twins has been presented to understand how physical settings can be modelled. From the above, it is clear that modelling means first recognising critical-to-solution (CTS) variables and then grabbing the hidden relationships between those variables that determine a given system behaviour (this latter can sometimes be confounded with the expected performances of the same system). Stated in this way, it is clear that the modelling effort becomes overwhelming by using model-based or, on the contrary, data-driven approaches, which are keen to work well once the decision maker clarifies the decisional scenario. The approaches developed so far to solve specific problems lack of a clear and systematic methodology to model and develop virtual models (DTs), i.e. to make explicit the embedded knowledge behind physical systems. In Object-oriented programming (OOP) [59], a pattern is defined as: *“a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over without ever doing it the same way twice”* [60]. Patterns in OOP help transfer the same solution among different implementations [61]. However, patterns in OOP are conceived and designed from experts.

This paper aims to prove that recognising recurring behaviours of systems thanks to patterns detected within data may significantly reduce the modelling complexity and make the decisional task of building the Digital Twin more objective. The expected results is a modelling paradigm in the form of invariant modelling patterns which allow more straightforward design, modelling, development, and hopefully maintenance of digital twins. Patterns have some trade-offs and limitations: experts are expected to develop and update the new pattern, i.e., this pattern-recognition process is not unmanned. Furthermore, the main consequence of using patterns is applying them across the same objects with the risk of applying too many unnecessary design patterns. In this context, the proper research challenge is thus how to identify automatically invariant modelling patterns within data, as schematised in Figure 2.

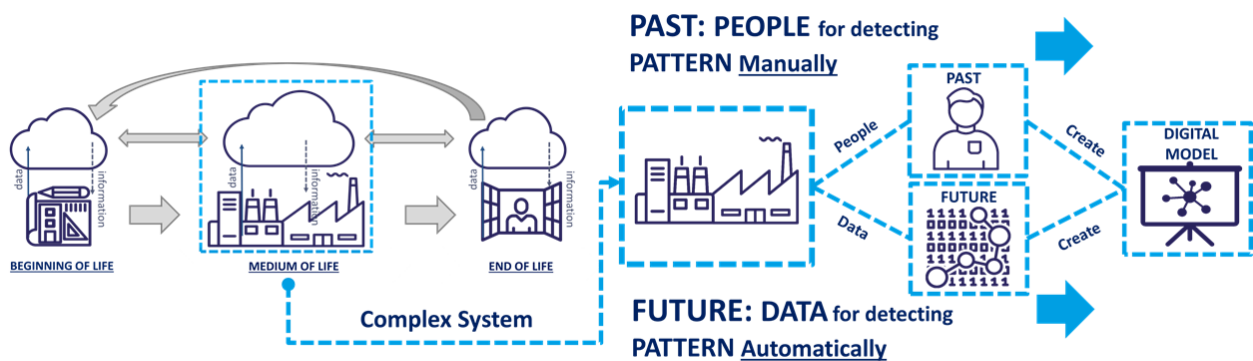


Figure 2: The research contribution

The invariance concept is the core of our proposed DT-design methodology, provided the capability to recognise invariant patterns means to identify a recurring structure within system components and thus to define “elementary behaviour” or a set of variables that never changes despite their instantiation. As consequence, it may therefore be then the case to create a library of patterns to form a sort of grammar of a helpful language to describe/represent the components of a model and then to build a DT.

In mathematics, invariance refers to a property of an object (say class) which remains unchanged after any operations, applied to the object [62]. In physics, invariance, also defined as symmetry, is the property possessed by an entity not to be modified by applying a transformation. Each property (also called symmetry properties) is always associated with conserving a physical quantity [63]. Scaling up the reasonings to a higher level of abstraction, we can refer to Wigner [64] speculating on the idea of symmetry and invariance principles stating that << *There is a strange hierarchy in our knowledge of the world around us. ... There is, nevertheless, a structure in the events around us, that is, correlations between the events of which we take cognisance*>>.

The idea proposed here is that knowledge behind the explicated correlations between CTS variables allows forecasting system behaviour and thus predicts the potential outcomes of uncertain events. To a certain extent, a DT is no more than a sophisticated model of reality, an implicit law embedding structured information aimed to perform the same scope: to predict the uncertain future evolution of a system, even the most difficult to foresee. In the context of building a DT for manufacturing systems, patterns are a means to recognise a feature or characteristic of a physical entity, as already stated. We can then extend the definition of invariance to a pattern in the context of building DTs: an invariant pattern describes/emulates an object/system independently of the context of the application of the system. Here we applied the concept of symmetry to patterns, provided a symmetrical pattern maintains the same features/attributes in different fields of application: a pattern will here be considered invariant as long as it has a behavioural symmetry over time or other contexts, thus leading to an unchanged behaviour at changing the reference domain (field of application). It is clear that symmetry will be a function of the quality and typology of the description of the system, more than of the specific system's features and state variables themselves.

Citing again [64], recognising invariance is thus the prerequisite to discovering events and advancing the world's knowledge. Invariance per sé thus is a value in building DT as it gives confidence in the model components' quality, provided these do not depend anymore on the specific situation. Invariant Modelling Patterns are thus useful in terms of i) identifying modelling constructs by explicating knowledge of the system's behaviour; ii) simplifying the modelling effort by allowing to transfer of an explicit feature (say system behaviour) among different modelling implementations, and iii) providing a weak proof of modelling quality and consistency thanks to recognising similarity in system's behaviour: the larger the similarities, the higher the consistency of the model. This latter point needs further speculation: manufacturing systems may be strongly different, but this is true depending on the point of view. Suppose one considers the physical aspects (i.e., logistics, materials, operational parameters, etc.), in this case, this difference appears evident, but if one considers the functional point of view (e.g., the satisfaction of needs) - at a higher level of abstraction- these differences may disappear. Patterns are those constructs able to capture functions (say, the behaviour of systems) and thus are susceptible to remain unchanged over time or space. It is then evident how the concept of "standardisation" of patterns (i.e. the capability of recognising classes or categories of patterns) may be an objective approach in building DT models to gain the confidence of the previously-mentioned model quality and consistency itself.

Another practical use of the idea of invariant modelling patterns is in terms of modelling efficiency since it allows reusable and parameterised modelling constructs corresponding to physical behaviours already recognised in advance to be easily re-applied in different situations. Through the invariant modelling patterns, the modelling time could thus be significantly reduced. Given the complexity of the reality of manufacturing systems, it is of significant interest to automate this pattern search to form the pattern library to support the modelling effort.

The practical methodology proposed in this paper starts with identifying patterns from historical data observed at multiple scales and levels. Accordingly, the detection and recognition of invariant modelling patterns are done by data-driven analysis. Patterns are here identified as characteristic and invariant structures or modules in data extracted from manufacturing systems. Invariant modelling patterns represent recurrent information or behaviour of a system that emerges as an outcome from historical data like the product, order, equipment, processing, and equipment information [65]. Once identified and made explicit, patterns provide a strong indication of the behaviour of a system and are typically described quantitatively. A pattern will thus represent an elementary unit (modules) of the modelling scheme that allows for building the overall virtual model of a given physical system or problem. The resulting library of patterns may automatically help to identify recurrent behaviours to "reproduce" real systems functioning and thus simulate and emulate it through a digital twin.

The advantages of invariant modelling methodology proposed will then be the reliability in modelling and the modularity, which facilitate the development and maintenance of digital twins. Modularisation, therefore, can improve DT modelling efficiency as it allows different patterns to be separated and recombined with the advantages of flexibility and reusability. This approach can dramatically reduce the workload and time required for developing DT. Based on patterns, the model building may result in just integrating selected constructs. At system change, the designer needs only to adjust or select the correct patterns, and the virtual DT model may hopefully be automatically rebuilt. We will show in the following sections how to detect, apply and especially re-use predefined functional patterns that are systematically developed and logically interlinked for the configuration of a holistic manufacturing system thanks to the industrial case discussed. The rest of the paper will be aimed at explaining and providing evidence of this idea of pattern standardisation as a novel approach to model DT: although it will not be formal proof, it will allow readers to speculate on the idea and draw their conclusions on this topic.

3. AUTOMATIC DEFINITION OF INVARIANT MODELLING PATTERNS

Identifying an invariant modelling pattern also requires the design of a methodological approach to recognise recurrent data structures and formalise their syntax and semantics. The further step presented here is a methodology for automatically detecting repeatable modelling patterns through data-driven analysis, which corresponds to typical behavioural characteristics of systems and, thus, derives solutions to commonly occurring modelling problems from this knowledge.

The methodological process of detecting invariant modelling patterns data-driven is shown in Figure 3 and consists of the following steps:

- (1) the first step is to identify the systems to analyse and the choice of products and the related manufacturing process;
- (2) A model-based approach aims to model the system by describing the function, the structure, and the behaviour;
- (3) the model-based approach thus draws a detailed representation of the system under consideration, enabling the selection of critical data to be collected and analysed;
- (4) when the system and the data type are selected, the data-driven approach can be applied to automatically detect and discover associations and relationships among data. Formal concept analysis (FCA) or Relation concept analysis (RCA) data-driven techniques are applied at this stage to analyse

manufacturing data and try to identify patterns under the shape of association rules. The associations can describe recurrent behaviours of the system and codify tacit knowledge that can be used to understand the system's behaviour better.

(5) The fifth step consists in analysing discovered associations based on the model developed in (2) to extract knowledge from data and define the physical meaning of the associations. In this way, invariant modelling patterns based on data-driven analyses can be detected;

(6) Patterns need to be formalised for use within different contexts with minimum effort.

(7) invariant modelling patterns can better help to design the virtual model, such as its corresponding digital twin. These can significantly advantage reusability among systems or processes operating in a similar condition.

According to the above, the proposed DT-modelling approach consists of eight different stages, put in a cycle as shown in Figure 3: 1) definition of the system; 2) system modelling; 3) data selection; 4) knowledge discovery; 5) knowledge extraction; 6) knowledge formalisation; 7) definition of design criteria for building a digital twin; 8) knowledge re-use. The first three steps are model-based oriented because these are functional to generally build the physics of the manufacturing system and select the right set of information to design a DT. Steps (4)-(8) are data-driven-oriented to access a sizeable amount of data to detect a priori all the possible invariant patterns behaviour in which performance degradation can occur. The following paragraphs explain every single stage in detail.

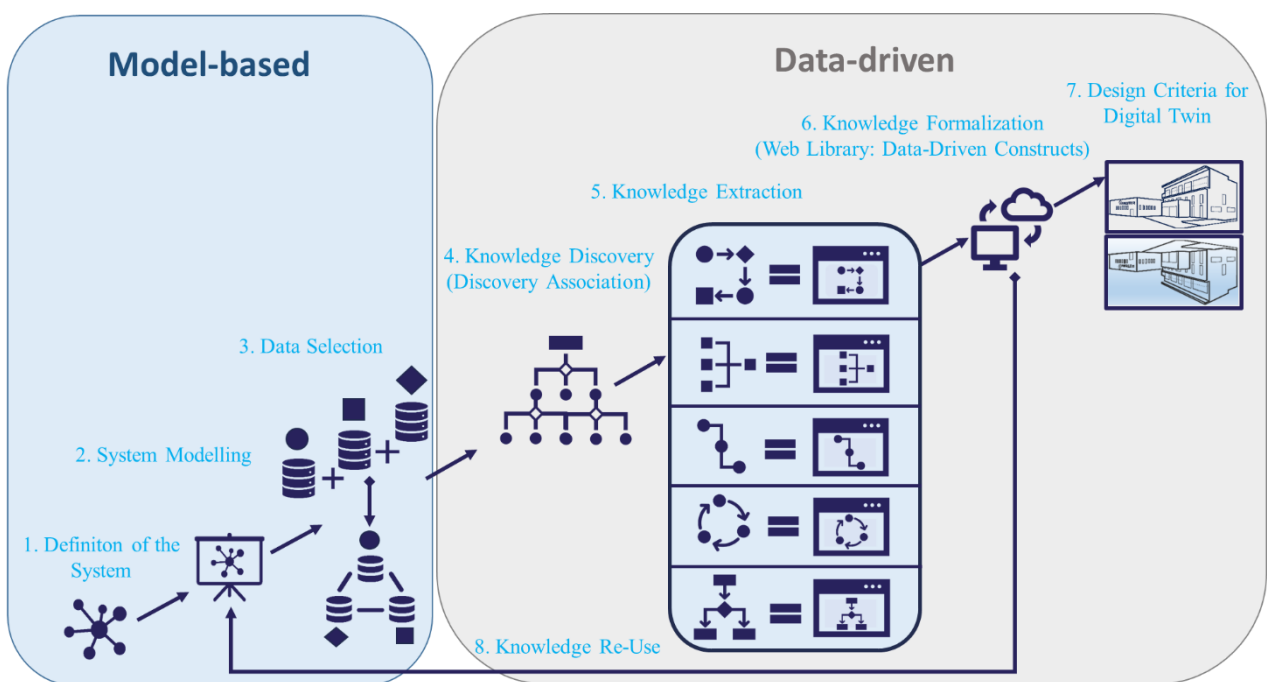


Figure 3: Invariant Modelling Patterns Methodology

3.1 DEFINITION OF THE SYSTEM

The first step for defining data-driven patterns is the definition of the system to be analysed by recognising the main variables and their interrelationships. For the definition of the system, it is necessary to identify the scenario under analysis. The approach for defining how to select a system is presented and described, as an example, in [66], [67], [68]. It consists of applying a thermodynamic model for selecting and designing a system. This thermodynamic view of the system represents a

guideline to structure the critical variables and the paths to digitalise processes for the intelligent control of the core production processes [67], [68].

3.2 SYSTEM MODELLING

System modelling aims to show how system components, their contents (Properties, Behaviours, Constraints), and their relationships interact [27]. MBSE approach is adopted in this stage using SysML as a modelling language. SysML is “*a general-purpose graphical modelling language for representing systems that may include combinations of hardware, software, data, people, facilities, and natural objects*”. SysML provides nine interrelated types of diagrams to reproduce and describe a system defining the requirements, the structure, and the behaviour for identifying the possible decisions or actions. The advantage of using SysML as a modelling standard is describing the syntax and semantics behind any system [69]. The modelling stages to be performed are described below and summarized in Figure 4:

- 1) Modelling requirements and use cases to represent system requirements provides a detailed description of a user's requirements and the system's functionality. A requirement specifies a capability or condition that must (or should) be satisfied, a function that a system must perform, or a condition a system must achieve. Requirement (req) and use case diagrams (uc) can be applied in this stage.
- 2) Modelling the system structure using the block definition diagram (bdd) to model a system in blocks defining the relationships between them, such as their hierarchical relationship. A block represents a modular structural unit that describes a system structure. It can define a type of logical or conceptual entity, a physical entity (e.g., a system), hardware, software, or data component. Blocks have different structural and behavioural features. The structural features of a block are the properties. Properties describe a block's structural aspects in terms of its relationship to other blocks and its quantifiable characteristics. Value properties describe quantifiable physical performance, and other block characteristics, such as weight or speed. Behavioural features declare the set of services that characterise the blocks.
- 3) Modelling the relationships between blocks in the internal definition diagram (ibd) to specify the internal structure of a single block. The ibd diagram provides an inner vision of every block in the block definition diagram to define the physical connection between two entities that communicate through an item flow.
- 4) Modelling constraints in the parametric diagrams (par). Constraints can correspond to any mathematical or logical expression, including time-varying expressions and differential equations. It can be re-used and bound with other constraints to represent complex sets of equations. A constraint block defines a set of parameters related to each other by a constraint expression. Parameters may have types, units, quantity kinds, and probability distributions. Constraint blocks can be defined in model libraries to facilitate specific types of analysis (performance, mass properties, thermal, etc.). The parametric diagram (par) shows how constraint properties are connected by binding their parameters to one another and the value properties of blocks.
- 5) Modelling the behaviour of the system in state machine diagrams (stm). A state represents a condition in the life of a block, and it is used to define which kind of change it undergoes in response to the occurrence of an event or a guard condition and what behaviours it performs. The state machine diagram is a dynamic behavioural diagram that shows the sequences

of states that an object or an interaction goes through during its lifetime in response to events. The state machine diagram simulates how the states change based on internal or external events. Typically, state machine diagrams describe the state-dependent behaviour of a block throughout its life cycle in terms of its states and the transitions between them.

SysML analyses the system requirements and the visual development of the system components, structure and behaviour. It is a graphical representation of the static structure and the dynamic behaviour of a system as it is.

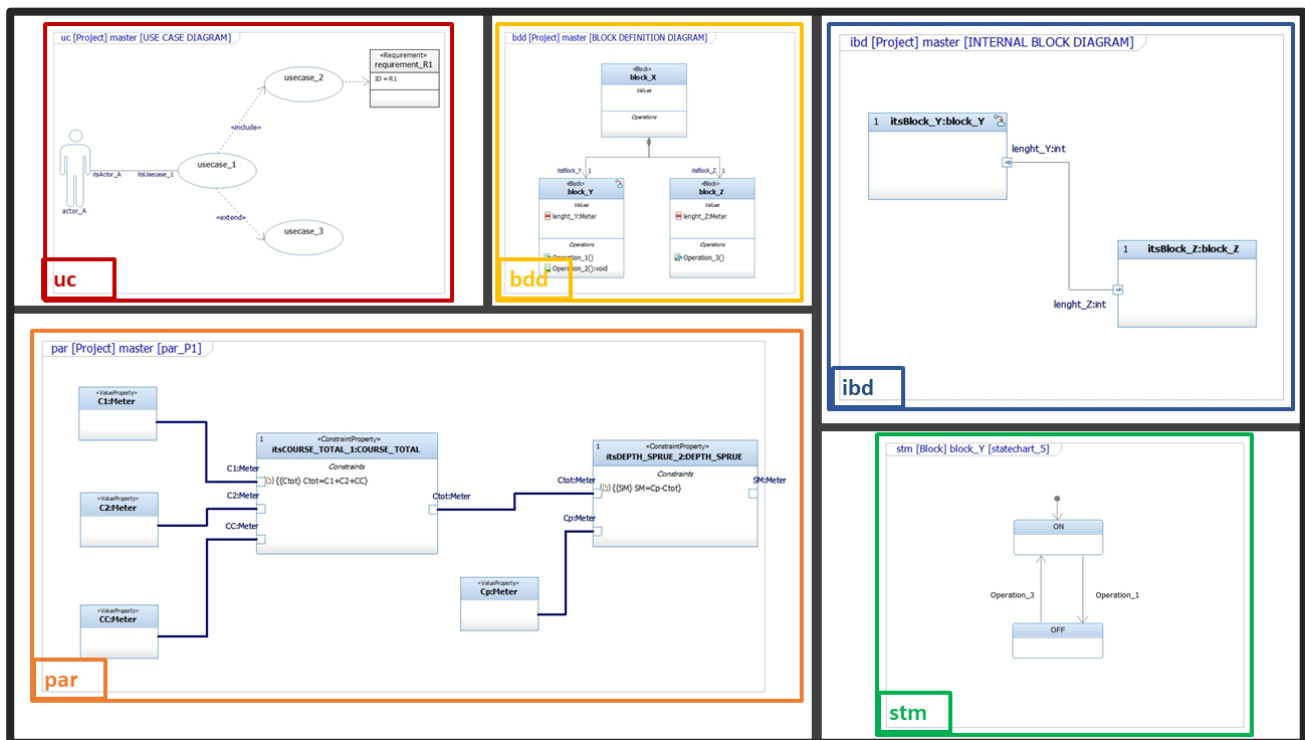


Figure 4: SysML Modelling Views

3.3 DATA SELECTION

The previous steps enable the model of the system, features, and performances, developing an understanding of the application domain and the relevant prior knowledge. The availability of knowledge about the processes allows for clearly tracking the system features and system evolution. The main idea is to select and collect data on which data-driven patterns discovery can be performed. Data selection is defined as the process of determining the appropriate data type and source and suitable instruments to collect data that includes a set of actions to complete. The first is creating a target data set, selecting a set, or focusing on a subset of variables or data samples. Second is data cleaning and pre-processing, where basic operations include removing noise, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, and accounting for time-sequence information and known changes. The third action comprises the merging of data from multiple data stores. This process must be carefully performed to avoid redundancies and inconsistencies in the resulting data set. Typical operations accomplished within the data integration are identifying and unifying variables and domains, the analysis of attribute correlation, duplicating tuples, and detecting conflicts in data values of different sources. Those actions enable collecting and converting data into information, sharing the information acquired,

formalising the knowledge, joint performance measurements, and leveraging the skills and knowledge [67].

3.4 KNOWLEDGE DISCOVERY: Data-driven analysis for detecting invariant modelling patterns

Formal concept analysis (FCA) or relation concept analysis (RCA) algorithms are applied at this level to detect data-driven patterns. FCA or RCA derive relationships and associations rules between objects described through attributes (e.g., operational parameters or manufacturing performance). The FCA is an unsupervised classification method oriented at knowledge representation and data-driven analysis [70]. This approach lets us elicit the conceptual structure of “object \times attribute” datasets [71]. It provides tools to group the data and to recognise formal patterns by representing them as a hierarchy of formal concepts (nodes) organised in a semi-ordered set named lattice (Wille, 2002), as shown in Figure 5. Relational concept analysis (RCA) [72] extends FCA to multi-relational datasets. RCA admits multiple objects in its input format, each organised as a separate context, plus a set of binary relations between contexts. RCA considers relations between objects from different contexts (tables) and builds a set of connected lattices. RCA will be revealing concepts that FCA –over the aggregated context with all possible nestlings of a depth up to the limit– will miss [73]. Relation concept analysis (RCA) extends FCA to processing multi-relational datasets, each providing attributes and relationships among them [72].

The FCA/RCA data table/tables consist of a set of objects (O) in rows and attributes (A) in columns, as shown in Table 2. Data are gathered within a (formal) context, a triple $K = (O, A, I)$, where O and A are non-empty sets, and I ($I \subseteq O \times A$) is a binary relationship [77], where $(o, a) \in I$, also written oIa , means that the object o bears the attribute a . The formal context (O, A, I) of an input matrix of n rows and m columns consists, as shown in Table 2, of a set of objects defined as $O = \{Obj_1, Obj_2, Obj_3, \dots, Obj_n\}$, a set of attributes defined as $A = \{Attr_1, Attr_2, Attr_3 \dots Attr_m\}$ and a binary relation I defined as $Obj_i, Attr_j \in I$ if and only if the intersection of i -th row and j -th column is not blank [78]. The symbol “**X**” indicates the existence of a relationship (I) between an object and an attribute, while the symbol “•” indicates the absence of it [79].

Table 2: FCA Formal Context, $K = (O, A, I)$ where $I (I \subseteq O \times A)$

| <i>O</i> | <i>A</i> | <i>Attr</i> ₁ | <i>Attr</i> ₂ | <i>Attr</i> ₃ | <i>Attr</i> ₄ | <i>Attr</i> ₅ |
|-------------------------|----------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| <i>Obj</i> ₁ | | X | • | X | • | • |
| <i>Obj</i> ₂ | | • | X | • | X | • |
| <i>Obj</i> ₃ | | • | X | X | • | X |
| <i>Obj</i> ₄ | | X | I | X | • | • |
| <i>Obj</i> ₅ | | • | • | X | • | • |
| <i>Obj</i> ₆ | | • | • | • | X | X |
| <i>Obj</i> ₇ | | • | X | X | • | • |

| | | | | | |
|---------|----------|----------|----------|----------|----------|
| Obj_8 | X | • | X | X | • |
| Obj_9 | X | X | X | X | X |

Given a set of objects (O), a set of attributes (A), and defined relations (I) between objects and attributes, a formal concept (node) represents a subset of objects (O) sharing the same subset of attributes (A), as displayed in Figure 5, where each node represents a concept. A concept is constituted by its extension, which consists of all objects belonging to the concept and its intention, which comprises all attributes shared by those objects. FCA/RCA algorithm converts the data tables automatically into connected concepts [75] to detect association rules between the objects and attributes from a single or different dataset [76], [77]. This understanding allows the formal discovery of association rules among concepts, consequently recognising which concepts are closely related based on the shared attributes [74].

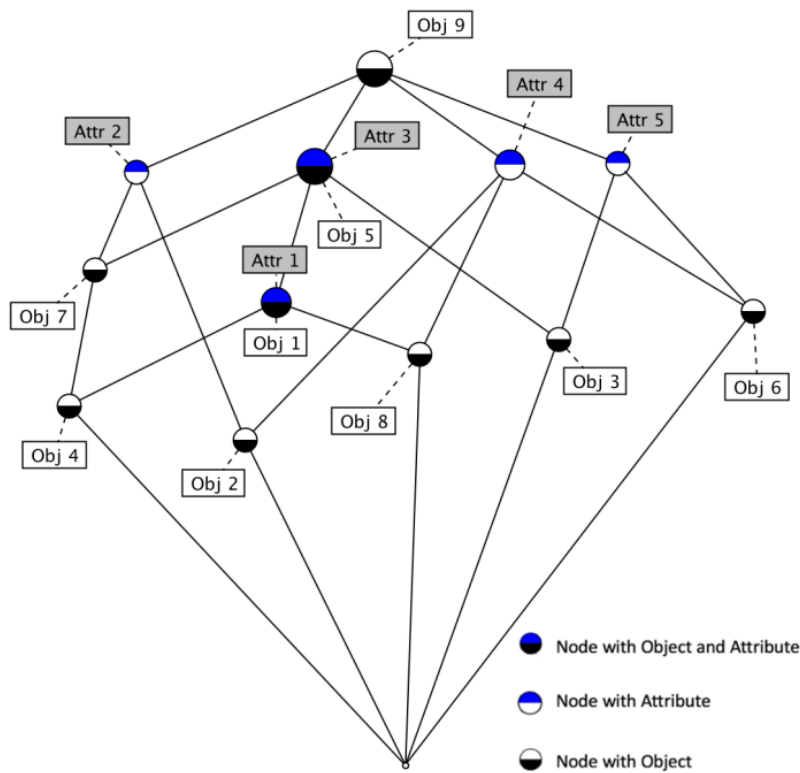


Figure 5: Example of Concept Lattice generated using FCA [80]

An association rule in FCA/RCA is a pair (X,Y) written as:

$$ID_Association.Rule: \langle o \rangle X \square \langle o \rangle Y [s,c] \quad (1)$$

It embodies information about co-occurrences of X and Y in objects from O where:

- ID_Association.Rule: Identification Number of a Concept.
- $\langle o \rangle$: the number of objects that share the same association rule.
- X is the association rule source (IF), i.e. the subset of objects sharing the same subset of attributes that generates Y.
- Y is the association rule target (THEN), i.e. the subset of attributes generated by X.
- Support (s) and confidence (c) are two classical association evaluation metrics.

The support (s) is the number of transactions that include items in the {X} as a percentage of the total transactions. It measures how frequently the collection of items occurs together as a percentage of all transactions.

$$\text{Support}(X) = \frac{\text{Number of rows(objects) in which } X \text{ appears}}{\text{Total number of rows(objects)}} \quad (2)$$

Confidence (c) measures how often each item in Y appears in transactions that contain items in X also.

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)} \quad (3)$$

For example, if we consider the attribute $Attr_1$, the attribute $Attr_2$ in Table 2 and the association rule ($Attr_1 \rightarrow Attr_2$), FCA/RCA calculate the support of $Attr_1$ and the confidence of the association rule as follows:

Association Rule: ($Attr_1 \rightarrow Attr_2$)

- $\text{Support}(Attr_1) = \frac{4}{9} = 0.4$
- $\text{Support}(Attr_2) = \frac{5}{9} = 0.55$
- $\text{Support}(Attr_1 \cup Attr_2) = \frac{2}{9} = 0.55$
- $\text{Confidence}(Attr_1 \rightarrow Attr_2) = \frac{\text{Support}(Attr_1 \cup Attr_2)}{\text{Support}(Attr_1)} = \frac{2/9}{4/9} = 0.2$

An association rule is very useful in analysing data and discovering patterns where a pattern is: “a formal structure knowledge about objects, by giving association rules between attributes describing the objects”. These latter are evaluated and selected based on the support and confidence values equal to 50% and 100% accordingly.

The RCA/FCA approach described above has been adapted herein to model a generic manufacturing system. A manufacturing system is “a collection of integrated equipment and human resources whose function is to perform one or more processing and/or assembly operations on a raw material, part or set of parts” [7]. It can thus be conceived, as shown in Figure 6, as a set of manufacturing transformations, where these latter correspond to the manufacturing system's operations. The inputs are the $n \geq 0$ input operating parameters to perform a given industrial transformation (that produces the final good). Performances measure the efficiency/efficacy of this transformation and thus can be schematised as outputs to our scopes (see Figure 6). A manufacturing system can then be conceived as a set of manufacturing transformations over time (say, the objects in table 2) and the operating parameters and the performance are the input and output, respectively (say, the attributes in table 2).



Figure 6: Logic schematisation of a Manufacturing Transformation for DT modelling purposes.

The RCA data-driven technique is more suitable for analysing a manufacturing system because it can simultaneously process and link input and output attributes. Our idea of RCA's application seems good to detect invariant modelling patterns for building DTs of a manufacturing system. The generic

RCA data structure is presented in Figure 7. To apply RCA to model the manufacturing system presented in Figure 6, it is necessary to create and process two different tables. The first table (D1) represents the input attributes, i.e. machined parts over time (objects O) and operating parameters (attributes A). The second table (D2) stores the output attributes, i.e. machined parts over time (objects O) and performance parameters (attributes A). The formal context (O, A, I) of table D1 consists, as shown in Figure 7, of a set of objects defined as $O = \{Obj_1, Obj_2, Obj_3, \dots, Obj_n\}$, and a set of attributes defined as $A = \{X_1, X_2, X_3 \dots X_m\}$. The formal context (O, A, I) of table D2 consists, as shown in Figure 7, of a set of objects defined as $O = \{Obj_1, Obj_2, Obj_3, \dots, Obj_n\}$, and a set of attributes defined as $A = \{Y_1, Y_2, Y_3 \dots Y_m\}$.

An invariant modelling pattern, detected through RCA, has the form:

$$ID_Association.Rule: \langle o \rangle X_1, X_m \square \langle o \rangle Y_m [s, c] \quad (4)$$

This represents a direct correlation and relationship between one or more operating parameters $\{X_1, X_2, X_3 \dots X_m\}$ with one or more performance $\{Y_1, Y_2, Y_3 \dots Y_m\}$. The confidence metric measures the association rule's accuracy between operating parameters and performances. Based on the support and confidence metrics, the association rule allows quantifying the existence of a relationship between parameters formally and thus detecting an invariant behaviour of any given element (say, process) in a manufacturing system. In particular, a pattern recognizes a possible performance behaviour based on a specific instantiation set of operating parameters. Thus, the association rules are strong proof of the existence of a generic relationship between parameters of a given manufacturing process, while the Support and Confidence metrics provide the instantiation of a given process.

Provided that our pattern-based approach to developing a Digital Twin relies on a general definition and use of patterns, we can put the following statement: once you have a set of patterns for any given system/process, it is possible to build an effective Digital Twin, whose reasoning will be based on the knowledge embedded into the set of patterns mentioned above. Optimisation problems, which are those typical problems we are facing in supporting decision-making using digital twins – at least in the manufacturing domain – are typically definable using patterns as a way of representing knowledge (i.e. relationships between process parameter /cause/ and related performances decay /effect/) helpful in getting to the solution. As an outcome of all of these reasonings, it is thus possible to model a digital twin of any manufacturing system once the existence of an invariant relationship between its parameters has been proved: performances will then be consistently recognised, modelled and predicted through the operating parameters identified by the domain expert for the specific domain (say manufacturing system). This latter point is still subject to the subjective decision of the expert, as in most of the present industrial applications.

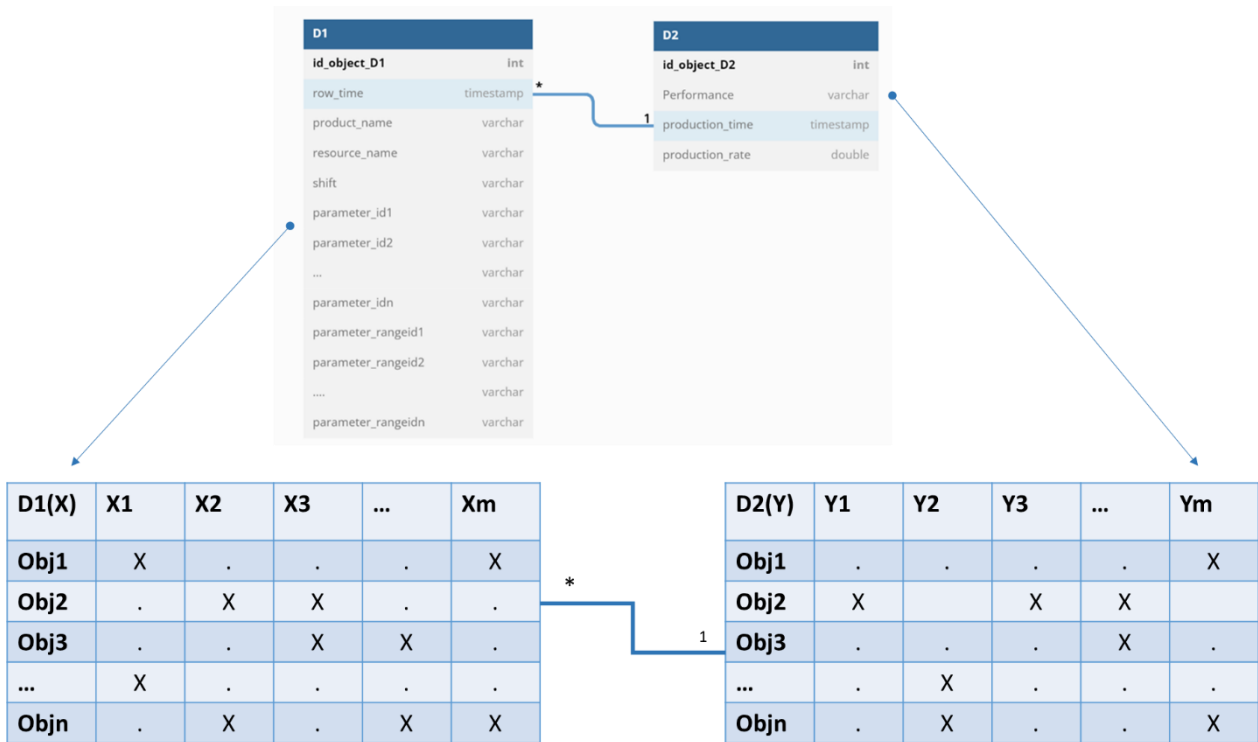


Figure 7: RCA design for a generic Manufacturing System

3.5 KNOWLEDGE EXTRACTION: Invariant Modelling Pattern Data-driven

This stage searches for data-driven patterns of interest in a particular representational form. The concept lattice supports knowledge extraction. The users can significantly detect patterns by correctly performing the preceding steps. Recognising a pattern and defining the correct mining is possible by evaluating it with the models performed in step 2. For analysis patterns to be reusable, they must be described concisely and consistently. We have devised the structure, shown in Figure 8, to formalize the patterns. The template lends a uniform structure to the information, making data-driven patterns easier to learn, compare, and use. This step also involves the visualisation of the extracted patterns and models. Each pattern is composed of six sections:

1. **ID PATTERN:** defines the univocal pattern ID identification.
2. **PATTERN NAME:** identifies the name and the function of the pattern.
3. **DESCRIPTION:** This short description answers the following questions: What does the data-driven pattern do? What design or modelling issue does it address?
4. **ASSOCIATION RULE (FCA/RCA):** this is an example of discovered associations among concepts based on FCA/RCA data-driven analysis.
5. **MODELLING VIEW (SysML):** this graphical SysML representation presents the modelling views defining properties, behaviours, and constraints. This section will help users understand the more abstract description of the pattern.
6. **APPLICABILITY:** defines how the pattern can be applied and defines how to recognise these situations. Concrete examples are also important because they help users choose which modelling solutions can be used.

| |
|---|
| 1. ID Pattern: unique name that helps in identifying the pattern (Pi) |
| 2. Pattern Name |
| 3. Description: a description of the goal behind the pattern and the reason for using it. |
| 4. Association Rule (FCA/RCA): an example of a concept for the pattern in analysis. |
| 5. Modelling View (SysML): a graphical representation of the association between data. Block definition, internal definition, parametric or state chart diagrams may be used for this purpose. |
| 6. Applicability: situations in which this pattern is usable. |

Figure 8: Invariant Modelling Pattern Structure

3.6 KNOWLEDGE FORMALISATION: WEB LIBRARY OF DATA-DRIVEN PATTERNS

The recognised knowledge needs to be formalised to be understandable and accessible to the users. Database management systems can store, collect, merge, and manage data from heterogeneous sources. Each pattern can be easily formalised, analysed, and applied.

3.7 DESIGN CRITERIA FOR MODELLING DIGITAL TWIN

The formalised data-driven patterns represent the design criteria for building efficient and straightforward end-user digital twins' interfaces to support the employees in decision-making. A data-driven pattern can include rules, constraints and deductions related to the production processes, such as the constraint of the processing capability of specific equipment. These can be formalised in algorithms to make the DT judge, evaluate, optimise and/or predict.

3.8 KNOWLEDGE RE-USE

The formalised knowledge can be used in other systems to design digital twins or to model other applications. In this way, data-driven patterns can be combined to create effortlessly dynamic models based on the specific application.

4. CASE STUDY: Die Casting Aluminium Process

The approach presented in section 3 has been tested for validation on a real manufacturing process of the company Master Italy s.r.l., an Italian SME that produces small hardware for civil window frames.

4.1 DEFINITION OF THE SYSTEM

The product and the processes are the steel corner and die-casting aluminium, according to the analysis presented in [68]. Die-casting aluminium is a manufacturing process in which molten metal

is poured or forced into steel moulds. The moulds, also known as tools or dies, are created using steel and are specially designed for each project. The total cycle time is short, typically around 33-35 seconds. The process cycle of die-casting aluminium consists of four main phases described below and summarized in Table 3, making explicit the technological modelling performed by an expert.

1. Melting: the aluminium enters the solid state and exits in the molten state.

Die-casting requires that aluminium is heated well into its liquid phase for injection. The melting point of aluminium is: 680-700 °C (T_{al}). Once melted and taken up to the proper temperature, the aluminium is transferred to each die-cast machine. Each die-cast machine has its holding furnace, which maintains the molten aluminium at temperature while waiting for use in the die-cast machine.

2. Injection: the molten aluminium is injected into the mould through a plunger.

The molten metal, maintained at a set temperature in the furnace, is then transferred into a chamber where it can be injected into the die. When a die-cast machine is ready for its next cycle (die is closed, ready for the shot), an automated ladle takes a prescribed volume of molten aluminium (V_{al}) from the holding furnace and pours it into the mould. Once pouring is complete, the injection phases begin. The first injection phase is slow (T_1) where the plunger moves forward at a low speed (V_1). After a prescribed distance (C_1), the plunger enters (C_2) an intermediate speed phase (V_2) where the speed is increased to fill the mould. Once this is complete (T_2), the machine enters a fast phase where speed is significantly increased to fill the part cavity with aluminium (CC)

3. Cooling: the molten aluminium solidifies in the mould cavity.

After the part cavity is filled and the plunger has stopped moving, the hydraulic cylinder pushing the plunger is pressurised to a higher pressure (PM). This pressure holds the molten metal in the dies during solidification. When the cavity is filled, and the molten metal solidifies, the final shape of the casting is formed. The die cannot be opened until the cooling time has elapsed and the casting is solidified. Clamping force (FC) must be applied to the die to keep it securely closed while the metal is injected. After a prescribed amount of time (TC), the die opens the ejector or moves half of the die.

4. Extraction: an ejection mechanism pushes the product out of the mould cavity.

Once the injection cycle is completed and the machine is fully open, the die-cast is pushed out, and the die-cast thickness (SM) is controlled to prevent quality defects. A die-cast represents an injection cycle. The die-cast contains 36 steel corners. It means 36 steel corners are produced for each injection cycle every 33 seconds or recast in case of a quality problem.

Table 3: Die-casting Aluminium Technological Parameters

| Die-cast Phases | Parameter | Unit | Definition |
|-----------------|-----------|----------------|--|
| Melting | T_{al} | °C | The melting point temperature of aluminium |
| Injection | V_{al} | m ³ | Volume of Aluminium |
| | T_1 | sec | Time of injection in the first phase |
| | V_1 | m/sec | Velocity of injection in the first phase |
| | C_1 | m | Course of injection in the first phase |
| | T_2 | sec | Time of injection in the second phase |
| | V_2 | m/sec | Velocity of injection in the second phase |
| | C_2 | m | Course of injection in the second phase |

| | | | |
|--|----|-----|---------------------|
| | CC | m | Multiplied course |
| | PM | Pa | Multiplied Pressure |
| | FC | N | Clamping Force |
| | TC | sec | Cycle-time |
| | SM | m | Die-cast thickness |

The expert classified the die-casting thermodynamic parameters upon a deep analysis as follows [68]:

- Main parameters already controlled: $V_{al}, T_{al}, T_f, C_1, C_2, CC, T_1, T_2, PM, FC, SM, TC$
- Main parameters not yet controlled are: $V_l, T_s, T_l, T_0, P_0, T_{mo}, P_{a1}, P_{a2}, P_{aM}$
- Derived parameters controlled are: V_1, V_2, PS, PF
- Non-controllable parameters are: $\rho_{al}, c_{p,al}, c_{l,al}, h_0, h_{al}, s_0, s_{al}$

The main controlled parameters are the parameters related to the volume and the temperature of the aluminium (V_{al}, T_{al}), the melting temperature of the aluminium (T_f), the course of the plunger in the first, second and multiplied phases of the injection stage (C_1, C_2, CC), the time of mould filling in the injection stage (T_1, T_2), the multiplied pressure (PM), the clamping force (FC), the thickness of the product (SM) and the cycle time (TC). The main parameters to control are, namely: the volume of aluminium loss during the injection stage (V_l), the temperature of the solid ingots at the entry of the furnace (T_s), the temperature loss during the injection stage (T_l), the environmental temperature and pressure (T_0, P_0), the temperature of the mould (T_{mo}), the pressure of the accumulator in the first, second and multiplied stage (P_{a1}, P_{a2}, P_{aM}). The derived parameters controlled are the speed in the first and the second phase of the injection stage (V_1, V_2) and the specific and final pressure (PS, PF). The non-controllable parameters are the density of aluminium (ρ_{al}), specific heat and latent heat of aluminium ($c_{p,al}, c_{l,al}$), specific enthalpy of environment and aluminium (h_0, h_{al}), specific entropy of the environment and the aluminium (s_0, s_{al}).

4.2 SYSTEM MODELLING

SysML has been applied according to the steps presented in section 3.2 to model 1) Structural composition, interconnection, and classification; 2) Constraints on the physical and performance properties; 3) Function-based and state-based behaviour; 4) Allocations between behaviour, structure, and constraints.

Firstly, the use case and requirements diagrams are designed, as shown in Figure 9, to depict some of the high-level functionality of the die-casting process using the SysML Rational Rhapsody software. The programmable logic controller (PLC) and the manufacturing execution system (MES) monitor the die-casting operating parameters and performance, respectively (say, D1 and D2 attributes in section 3.4). PLC is an industrial digital computer that monitors the technological parameters listed in Table 3. The manufacturing Execution System (MES) is the information system that monitors performance, i.e. productivity and machine downtimes. For this reason, ‘Monitor technological parameters’ and ‘Monitor machine downtime’ are defined as use cases for the process under analysis. The correlation between operating/technological parameters and machine downtime (performance) allows a better understanding of the system's behaviour for building the predictive behaviour models of the die-casting Digital Twin. Thereby, ‘Correlate technological parameters to machine downtime’

is added as an included use case to refine the requirement: ‘Predictive manufacturing’, i.e. building a DT for the die-casting aluminium process to predict and prevent all possible failures.

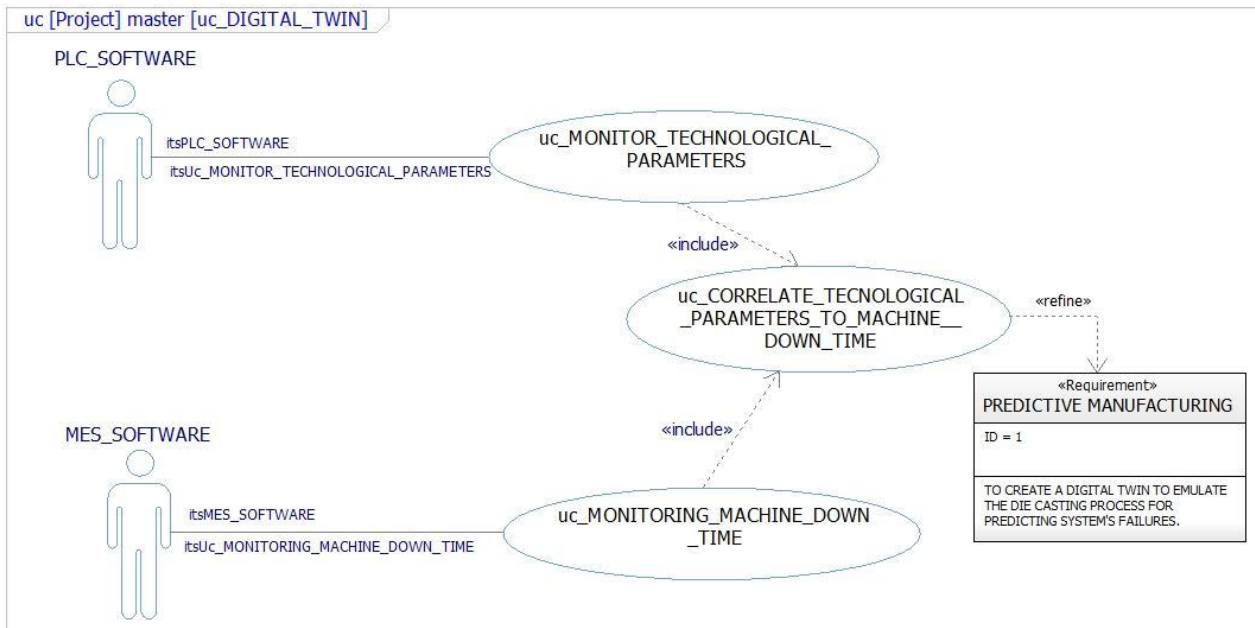


Figure 9: Use Case Diagram

Secondly, the die-casting aluminium process is modelled in a block definition diagram (bdd) in Figure 10, composed of four blocks representing the die-casting process phases: melting, injection, cooling, and extraction described in section 4.1 and Table 3. Each block is also described by a set of value properties, i.e., a quantifiable property with units, dimensions, and probability distribution. For example, the melting subsystem is a block characterised by three value properties, namely aluminium temperature (T_{al}), melting temperature (T_f), and aluminium temperature at the solid state (T_s) with a unit of Celsius. Constraint blocks are modelled on the same diagram to define the main thermodynamic and physical equations associated with each block's value properties.



Figure 10: Block Definition Diagram of Die-casting Process (bdd)

A second block definition diagram is designed in Figure 11 to the list of parameters monitored and controlled on-site by the programmable logic controller (PLC) and the manufacturing execution system (MES).

The programmable logic controller (PLC) monitors and records the run-time of the following operating parameters: the univocal ID of the injection cycle (MEASSETID), the name of the article to product (ARTICLECODE), the name of the machine (RESCOURCENAME), data and time for each MEASSETID (TIMESTAMPLOCAL), the number of injection cycle for a single production order (N_INJECTION). Furthermore, it monitors and records the following operating parameters: the course

of the plunger in the first phase (C_1) and in the second phase (C_2) of the injection stage, the time of injection in the first (T_1) and in the second phase (T_2), the speed in the first (V_1) and in the second phase (V_2) of the injection, the multiplied course (CC), the multiplied pressure (PM), the specific pressure (PS), the final pressure (PF), the clamping force (FC), the thickness of the product (SM) and the cycle time (TC). Each parameter has a minimum value (MIN) and a maximum value (MAX) that are listed as value properties, in the block PLC, as follows: MIN_C1, MIN_T1, MIN_V1, MIN_C2, MIN_T2, MIN_V2, MIN_CC, MIN_PM, MIN_PF, MIN_PS, MIN_FC, MIN_SM, MIN_TC, MAX_C1, MAX_T1, MAX_V1, MAX_C2, MAX_T2, MAX_V2, MAX_CC, MAX_PM, MAX_PF, MAX_PS, MAX_FC, MAX_SM, MAX_TC. The PLC automatically generates alarms if a machine malfunction occurs based on the minimum and maximum values.

The MES system controls multiple elements of the production process (e.g. inputs, personnel, machines and support services) to monitor the production output and performances. The parameters listed as value properties in the block MES, are univocal ID (ID_RKD), day of production (DAY_SHIFT), day and time (DATA_TIME), a possible state of the machine (STATE), possible activities of the machine (ACTIVITIES), name of the machine (RESOURCE), name of the employee (WORKER), the ID of the production order (PRODUCTION_ORDER), name of articles to product (ARTICLE), duration time of a setup (SETUP_TIME), duration time of production (UPTIME), duration time of maintenance downtime (MAINTENANCE_TIME), duration of a downtime (DOWNTIME), the quantity of conformed products (CONFORMED_QUANTITY), the quantity of discarded products (DISCARDED_QUANTITY), number of injection cycles during the production time (N_MOULDED), the univocal ID of a machine downtime (ID_MACHINE_DOWN_TIME), description of a possible machine downtime (DESCRIPTION_MACHINE_DOWN_TIME), shift (morning/afternoon/evening) (N_SHIFT), number of products for each injection cycle (N_PIECES).

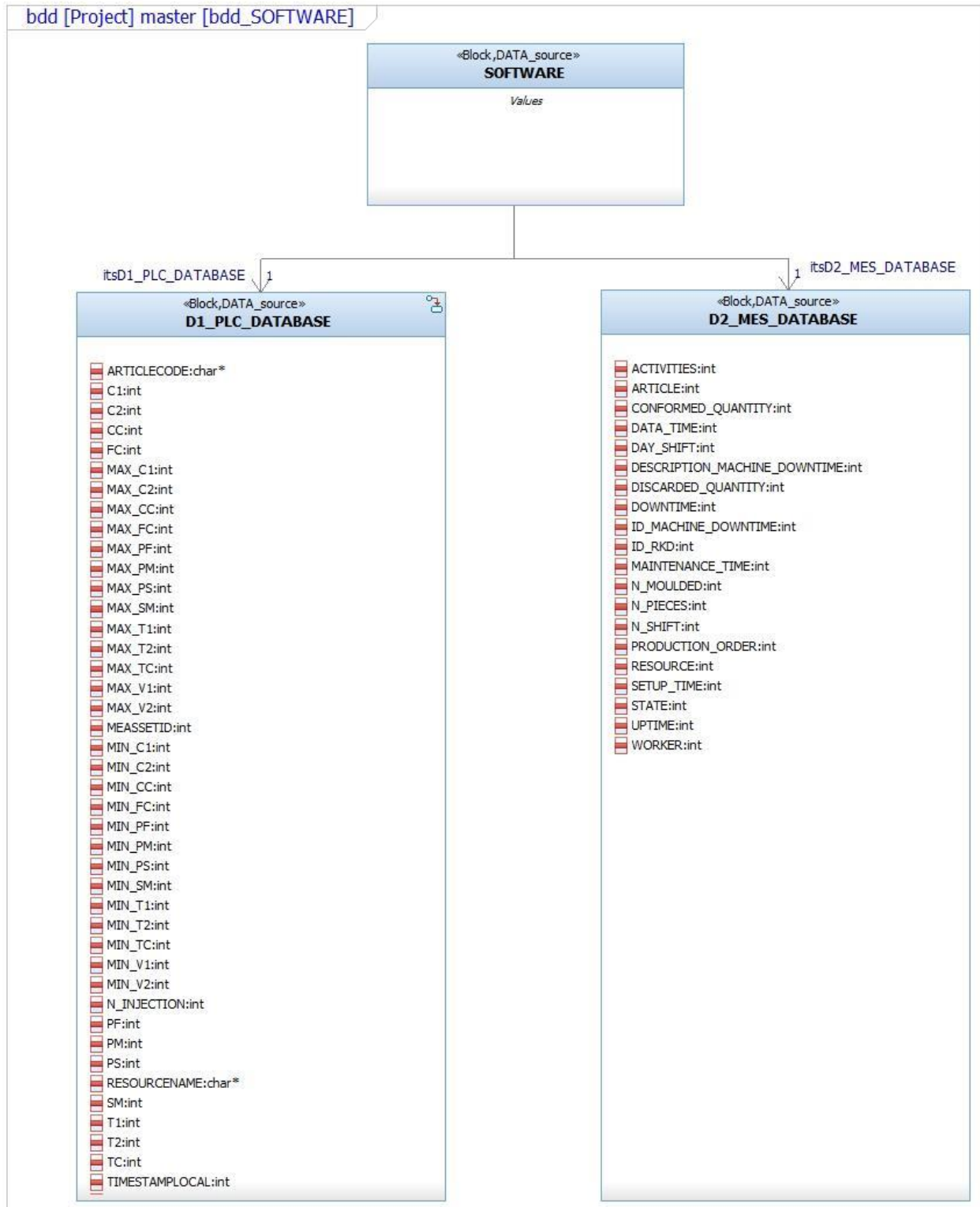


Figure 11: Block Definition Diagram (bdd)

Thirdly, the internal block diagram (ibd) is designed in Figure 12, showing how the two-block definition diagrams (Figures 11-12) presented above interact. The system starts with a data acquisition system using appropriate sensors to record signals such as course, pressure, speed, temperature, etc. The programmable logic controller (PLC) stores the values of the technological parameters from connected sensors or input devices. It processes the data and triggers outputs based on pre-programmed parameters (MIN and MAX values for each parameter) to monitor the injection,

cooling and extraction phases. The manufacturing execution system (MES) tracks production data to monitor the process's performance and productivity, recording all process downtime.

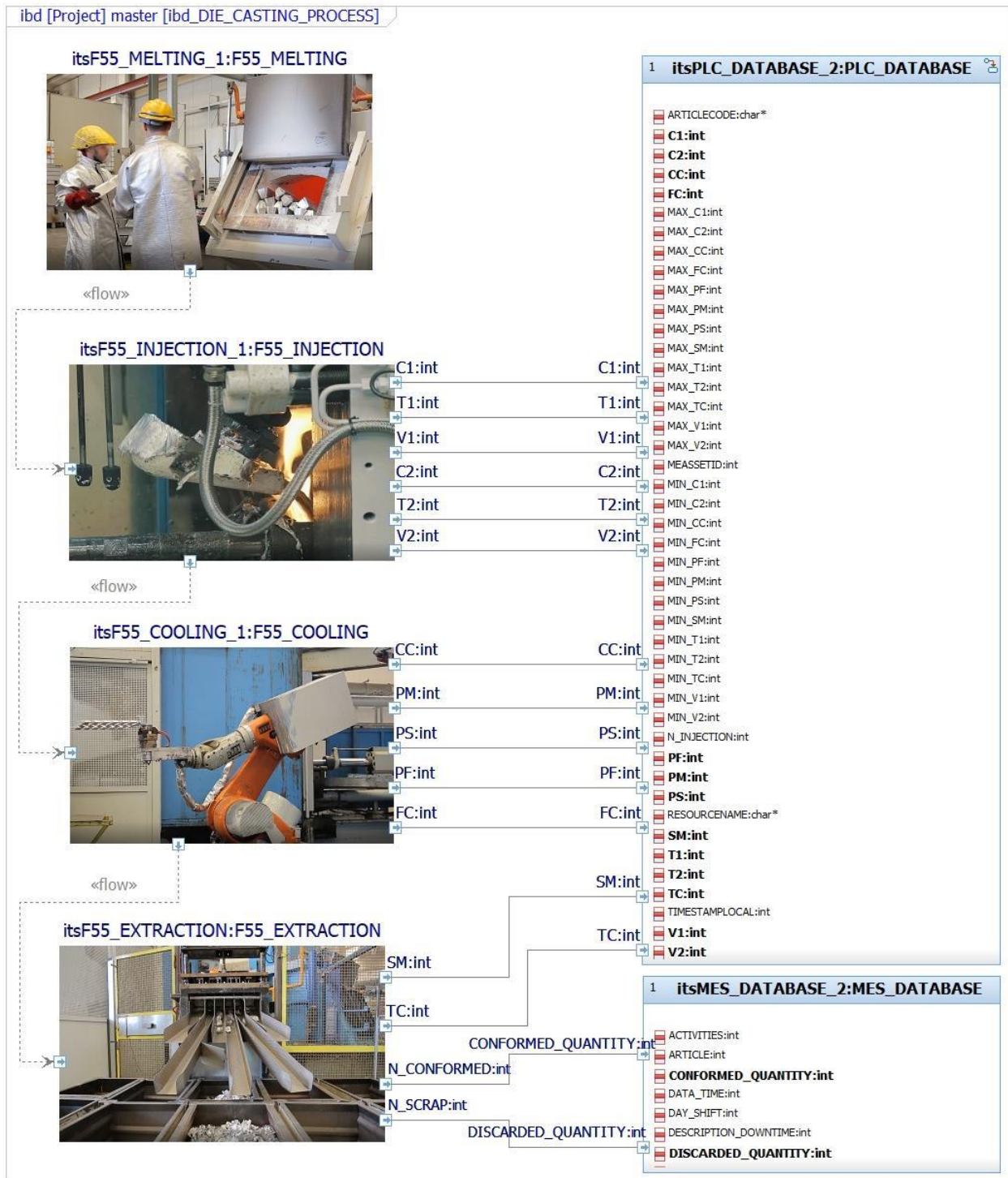


Figure 12: Internal Block Definition Diagram (ibd)

Fourthly, the parametric diagram in Figure 13 is created to model equations defined in the constraint blocks in Figure 8. For the sake of conciseness and vice versa, only a part of the die-casting equations is shown in Figure 13 to display the bindings between constraint parameters in different constraint expressions to create a composite system of equations.

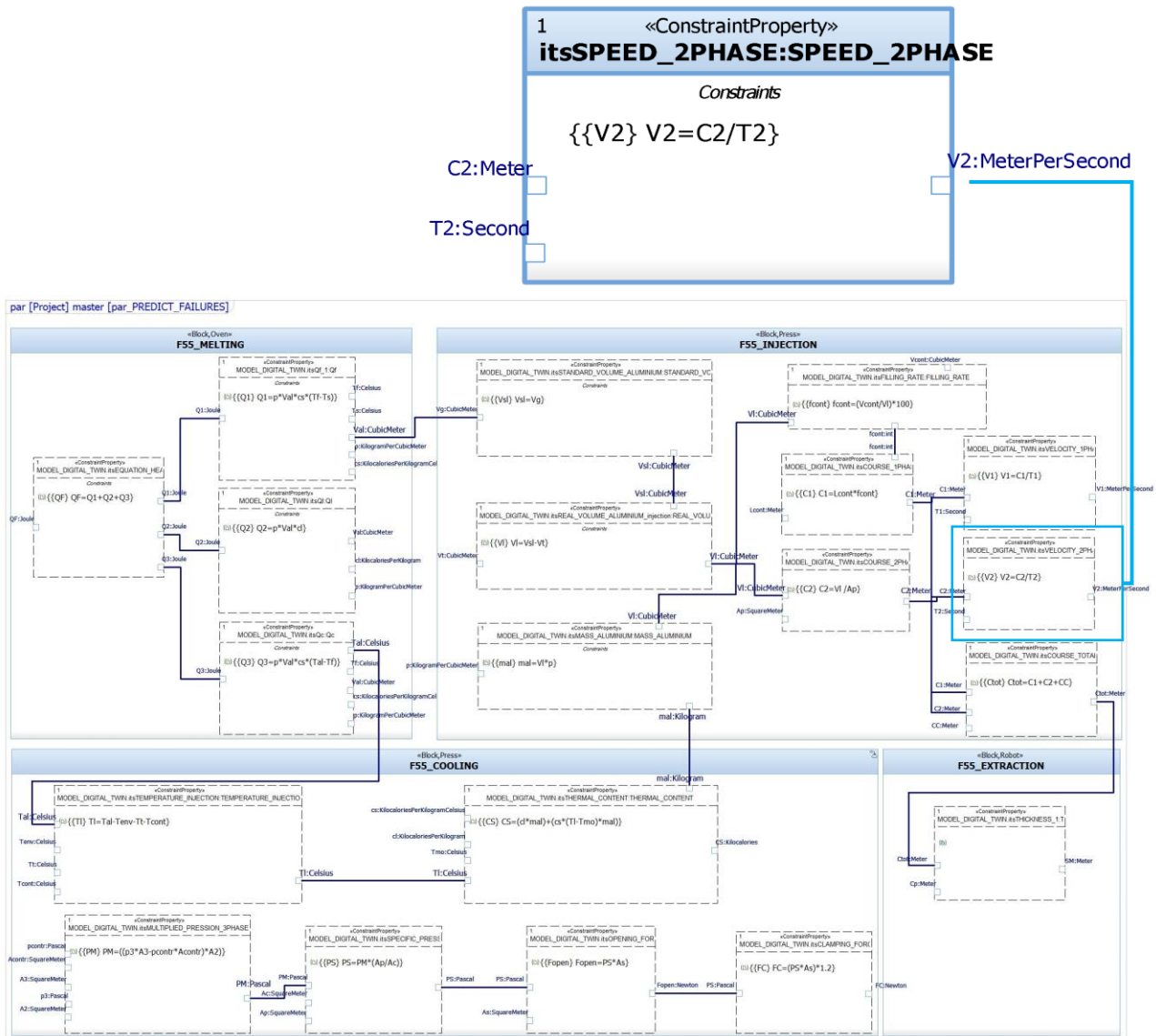


Figure 13: Parametric Diagram (par)

Lastly, the process's dynamic behaviour is expressed in the state machine diagram (stm) in Figure 14. The state diagram reproduces how the PLC records and evaluates the parameters defined in Figure 11, recognising four different states (see Table 4):

- **RECASTING (red)** when a PLC parameter is out of range, it causes the recasting of the injection cycle (≈ 36 products). V_2 , PM, FC, SM, and TC are the critical parameters because if V_2 , PM, FC, SM, and TC are equal to or below the minimum value or greater than or equal to the maximum value, there is the recasting of the injection cycle. A recasting is recognised as scrap (DISCARDED_QUANTITY) by the MES.
- **NOT CONFORMED (burgundy)**: when a PLC parameter is out of range, but it does not generate the recasting of the injection cycle.
- **CHECK (yellow)**: when a PLC parameter is close out of range.
- **CONFORM (green)**: when a PLC parameter operates into the defined range.

The numbers 1-5 define five different ranges of values where each parameter can operate based on the machine setup.

Table 4: PLC Parameters States Example

| PARAMETER | DATA TYPES | RANGE | STATE (S) | S |
|------------------------|------------|-------|----------------------|--------|
| C1- Course First Phase | NUMBER(10) | C1_1 | NOT CONFORMED | Red |
| C1- Course First Phase | NUMBER(10) | C1_2 | CHECK | Yellow |
| C1- Course First Phase | NUMBER(10) | C1_3 | CONFORMED | Green |
| C1- Course First Phase | NUMBER(10) | C1_4 | CHECK | Yellow |
| C1- Course First Phase | NUMBER(10) | C1_5 | NOT CONFORMED | Red |
| | | | | |
| V2- Speed Second Phase | NUMBER(10) | V2_1 | RECASTING | Red |
| V2- Speed Second Phase | NUMBER(10) | V2_2 | CHECK | Yellow |
| V2- Speed Second Phase | NUMBER(10) | V2_3 | CONFORMED | Green |
| V2- Speed Second Phase | NUMBER(10) | V2_4 | CHECK | Yellow |
| V2- Speed Second Phase | NUMBER(10) | V2_5 | RECASTING | Red |
| | | | | |

The state chart diagram in Figure 14 models the dynamic behaviour of the die-casting aluminium. The state machine diagram shows the possible state of each parameter and the events that can trigger a transition between the states. The process is initiated in the ‘OFF’ state. When it is ready to be evaluated, the ‘EvOn’ event triggers a transition from the ‘ON’ state to ‘READ’ state, as shown in Figure 14a. Upon entry to the ‘READ’ state, a timed event triggers a transition to ‘VERIFY_C1’ state in a time interval of 3 seconds. Once the process has entered in ‘VERIFY_C1’ state, it immediately transitions to the neutral state. A time event of 2 seconds triggers a transition to the condition connectors. It splits a single segment into several branches. Branches are labelled with guard conditions that are evaluated contemporary to determine which branch satisfies the condition. The connector evaluates if the value $C1=150$ activates the state C1_1 or C1_2 or C1_3 or C1_4 or C1_5 (Figure 14a). A time event of 3 seconds triggers a transition to the ‘VERIFY_T1’ state. The logic is the same as the one just presented. The guard conditions for the ‘VERIFY_V2’ state in Figure 14b are evaluated based on the constraint $(V_2 = \frac{C_2}{T_2})$ modelled in the parametric diagram in Figure 13. If the state V2_1 or V2_5 are activated, the number of DISCARDED_QUANTITY is calculated as follows:

$$tm(1000)/setDISCARDED_QUANTITY(DISCARDED_QUANTITY +1 \tag{5}$$

When also the ‘VERIFY_SM’ state is evaluated as shown in Figure 14c, the presence of a fault or a machine downtime should be detected. It requires the knowledge of all possible correlations between technological parameters with machine downtime for predicting all possible faults or breakdowns. For this reason, the RCA application aims to detect invariant modelling patterns in the knowledge discovery and extraction stages.

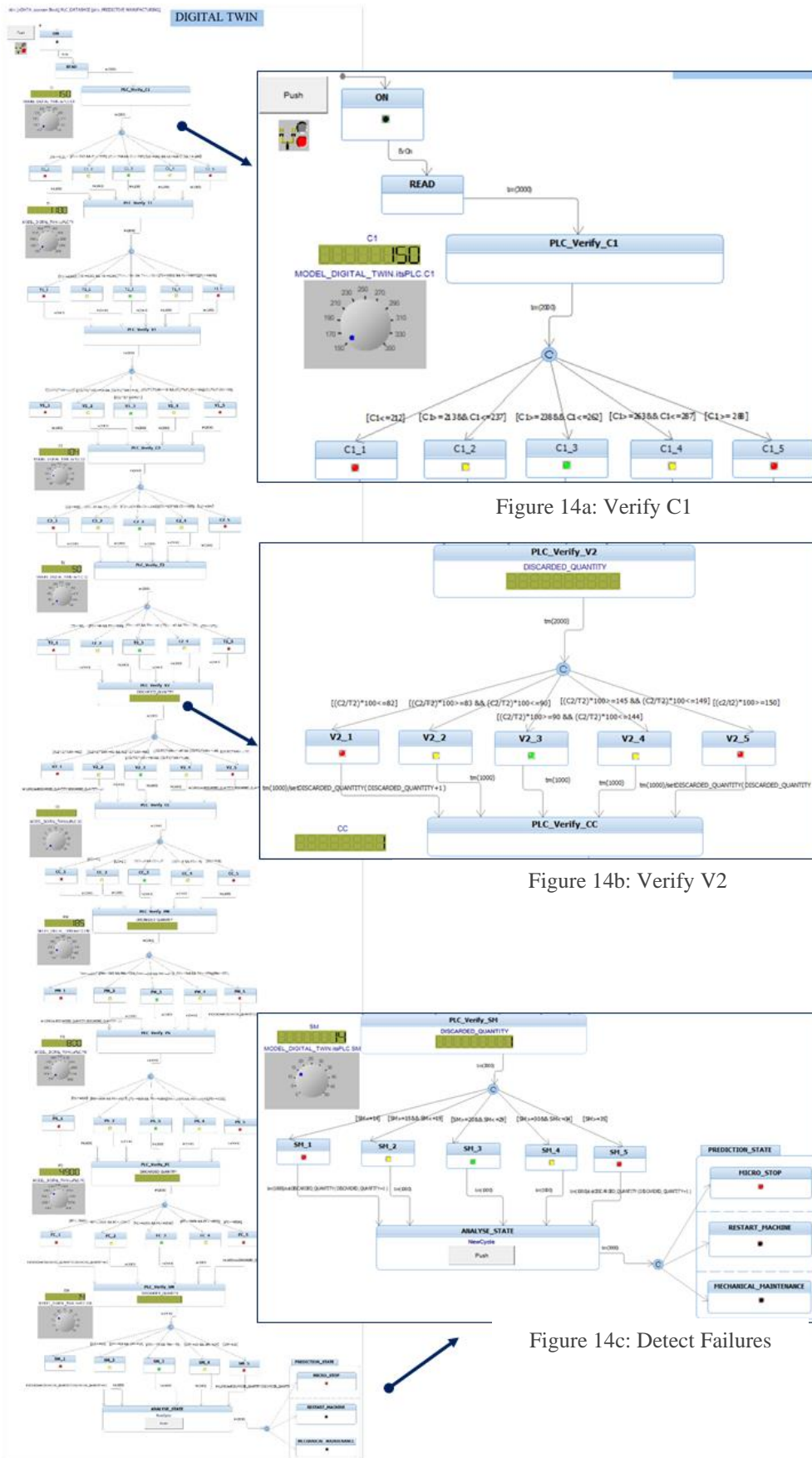


Figure 14a: Verify C1

Figure 14b: Verify V2

Figure 14c: Detect Failures

Figure 14: State Chart Diagram (stm)

4.3 DATA SELECTION

Two databases have been selected and analysed according to the logic explained in section 3.4 and steps 4.1 and 4.2. As described in section 3.4, the first table (D1-X) represents the input attributes of a manufacturing system, i.e. machined parts over time (objects O) and operating parameters (attributes A). The second table (D2-Y) stores the output attributes, i.e. machined parts over time (objects O) and performance parameters (attributes A). For the manufacturing system under analysis (Figure 15), the operating parameters are collected and stored in the PLC machine (see Table 3 and Figures 10-11), while the performances are measured by the Manufacturing execution system (MES - see Figure 11). In this case study, the operating parameters (X_i) are the technological parameters of a die-casting aluminium process, while the performance (Y_i) are assessed as machine downtimes. PLC data need to be gathered and combined with MES data to define all possible and existing correlations between technological parameters (X_i) and machine downtime (Y_i). This aims to discover explicit and tacit functional patterns for modelling the behaviour of a DT able to optimize the die-casting process's efficiency (i.e. prediction or reduction of downtime, waste reduction, maintenance and quality optimization).

D1(X). PLC: Operating Parameters= Technological Parameters

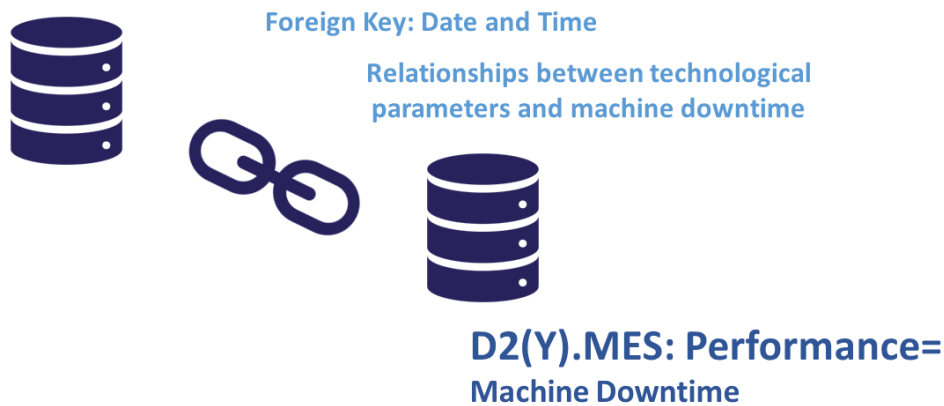


Figure 15: D1 and D2 Relationships for die-casting aluminium process

4.4 KNOWLEDGE DISCOVERY: Data-driven analysis for detecting invariant modelling patterns

The PLC and MES databases cover the last 18 months of production. D1 table (PLC) has 57.779 objects (O) and 67 attributes (A). D1 objects (OBi_{D1}) are the injection cycle. D1 attributes (ATi_{D1}) are PLC technological parameters listed in Table 3 (say, input attributes in section 3.4). D2 table (MES) has 57.779 objects (O) and 19 attributes (A). D2 objects ($OBi_{D2} = OBi_{D1}$) are the injection cycles. D2 attributes (ATi_{D2}) are the machine downtime problems listed in Table 6 (say, output attributes in section 3.4). RCA runs on two different tables, as shown in Figure 16. The objects and attributes of each RCA table are listed in Tables 5 and 6. The outcome is the lattice shown in Figure 17 that allows the detection of all possible patterns between the technological parameters and machine downtime.

D1(PLC) has 57.779 objects (G) and 67 attributes (M).

The objects (OBi_{D1}) are the injection cycle.

The attributes (ATi_{D1}) are the technological parameters.

| D1(PLC) | $AT1_{D1}$ | $AT2_{D1}$ | $AT3_{D1}$ | ATi_{D1} |
|------------|------------|------------|------------|------------|
| $OB1_{D1}$ | X | . | X | . |
| $OB2_{D1}$ | . | X | . | . |
| $OB3_{D1}$ | . | . | X | X |
| OBi_{D1} | X | . | X | X |

D2 (MES) has 57.779 objects (rows) and 19 attributes (columns).

- The objects (OBi_{D1}) are the injection cycles.
- The attributes (ATi_{D2}) are the machine downtime problems.

| D2(MES) | $AT1_{D2}$ | $AT2_{D2}$ | $AT3_{D2}$ | ATi_{D2} |
|------------|------------|------------|------------|------------|
| $OB1_{D1}$ | . | X | . | X |
| $OB2_{D1}$ | X | . | . | X |
| $OB3_{D1}$ | X | . | . | . |
| OBi_{D1} | . | X | X | . |

Figure 16: RCA Data Structure

Table 5: RCA Objects (O) of each Database (D1, D2)

| PLC (D1) OBJECTS (OBi_{D1}) | MES (D2) OBJECTS ($OBi_{D2} = OBi_{D1}$) |
|--|---|
| <p>The data table of PLC has 57.779 objects. The objects (OBi_{D1}) are the injection cycle. A row is generated every 30 seconds.</p> | <p>The objects (OBi_{D2}) are the injection cycle.</p> |

Table 6: RCA Attributes (A) of each Database (D1, D2, D3)

| PLC (D1) ATTRIBUTES (ATi _{D1}) | MES (D2) ATTRIBUTES (ATi _{D2}) |
|--|--|
| <ul style="list-style-type: none"> • A_M, B_P, C_N to define the shift (A: morning; B: afternoon; C: night). • Product_ID • C1_1, C1_2, C1_3, C1_4, C1_5, C2_1, C2_2, C2_3, C2_4, C2_5, T1_1, T1_2, T1_3, T1_4, T1_5, V1_1, V1_2, V1_3, V1_4, V1_5, T2_1, T2_2, T2_3, T2_4, T2_5, V2_1, V2_2, V2_3, V2_4, V2_5, CC_1, CC_2, CC_3, CC_4, CC_5, PM_1, PM_2, PM_3, PM_4, PM_5, FC_1, FC_2, FC_3, FC_4, FC_5, PS_1, PS_2, PS_3, PS_4, PS_5, SM_1, SM_2, SM_3, SM_4, SM_5 to define the technological parameters and their corresponding range. • OK, KO, CHECK to define the quality state of the injection cycle. | <ol style="list-style-type: none"> 1. Thickness SM<=14 2. Thickness SM=15 3. Thickness SM=16 4. Micro stop for SM_LimLow 5. Component problem 6. Electrical maintenance 7. Mechanical maintenance 8. Recasting start machine for micro stop (after SM_LimLow) 9. Recasting for SM_LimUpp 10. Recasting start machine for generic micro stop 11. Recasting start machine after cleaning of the equipment 12. Recasting start machine after auto-maintenance 13. Recasting start machine after component problem 14. Recasting start machine after electrical maintenance 15. Recasting start machine after mechanical maintenance 16. Recasting start machine after operator absence 17. Recasting start machine after not planned 18. Recasting start machine after setup 19. Recasting start machine after mechanical control |

4.5 KNOWLEDGE EXTRACTION

Applying the RCA, 7.743 association rules (possible patterns) have been detected automatically using the INCLOSE IV algorithm [76]. Here, we focus on 50% support (s) and 100% confidence (c) rules. The RCA lattice is shown in Figure 17.

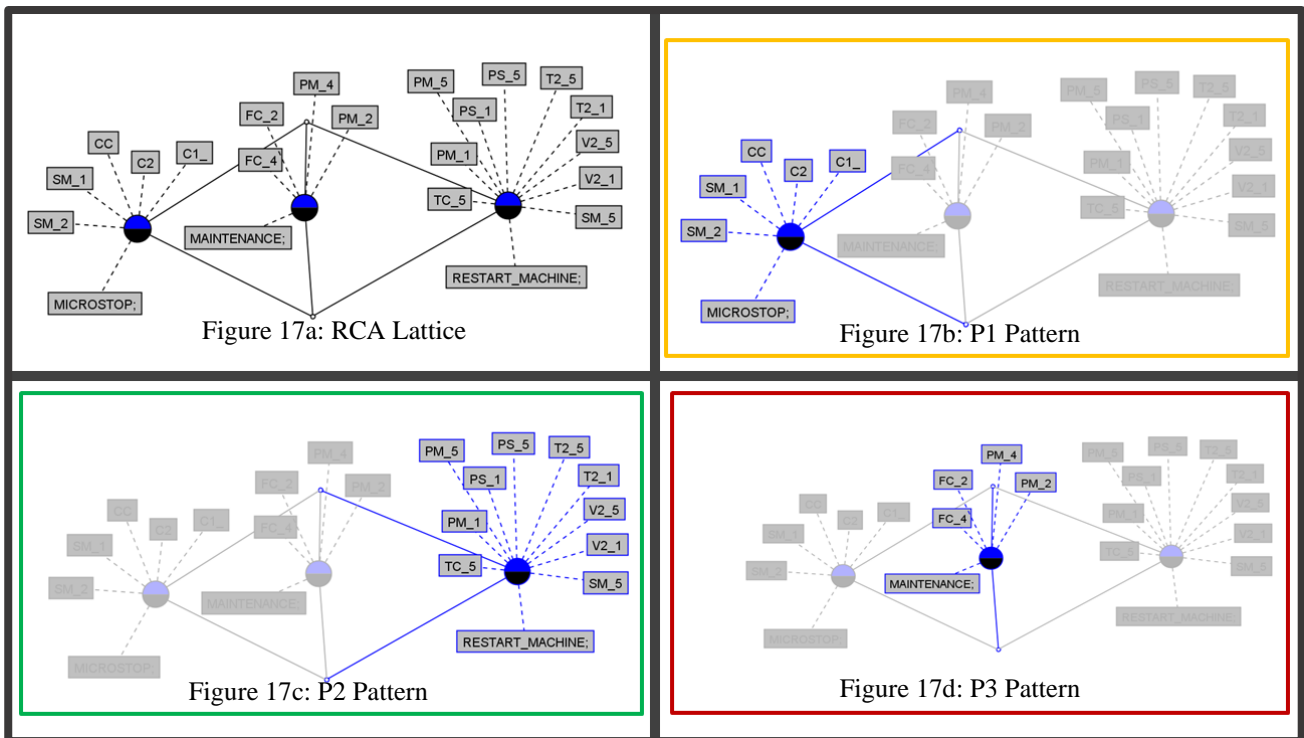


Figure 17: RCA Lattice and detected Patterns (P1,P2,P3)

The RCA lattice analysis (Figure 17) enabled the identification of three main invariant modelling patterns represented in Figure 18 and described as follows:

- P1** The downtime problems called: Thickness =16, Thickness =15, Thickness <=14, Micro-stop for SM_LimLow (say, problems 1-4 in Table 6) are caused by the following technological parameters: C_1 , C_2 , CC, SM_1, SM_2.
- P2** All recasting problems (say, problems 8-19 in Table 6) are associated with the following parameters: SM_5, V_{2_1} , V_{2_5} , T_{2_1} , T_{2_5} , PS_1, PS_5, PM_1, PM_5, TC_5.
- P3** The machine downtime related to the component problem, electrical maintenance and mechanical maintenance (say, problems 5-7 in Table 6) is generated by PM_2, PM_4, FC_2, and FC_4 parameters.

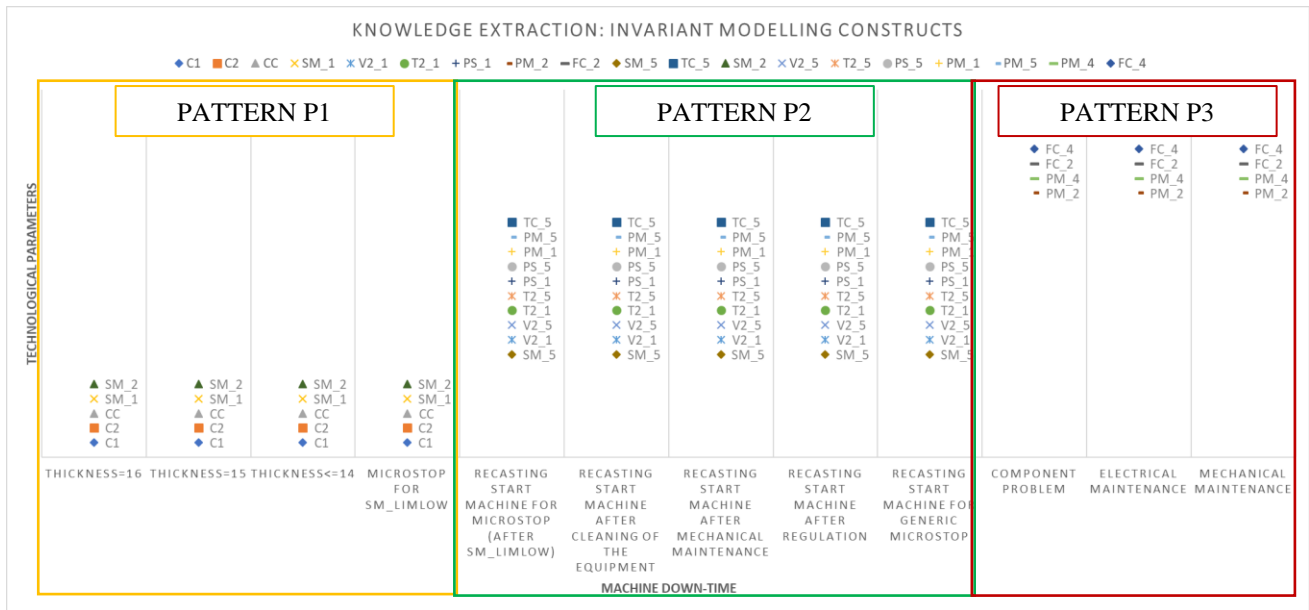
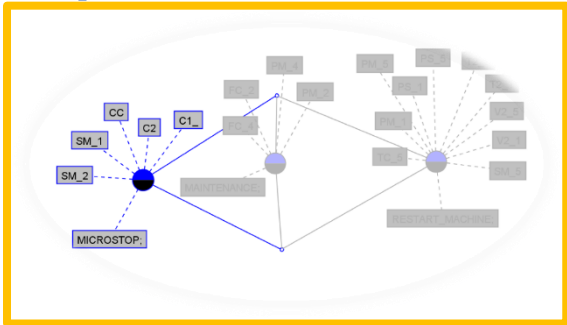
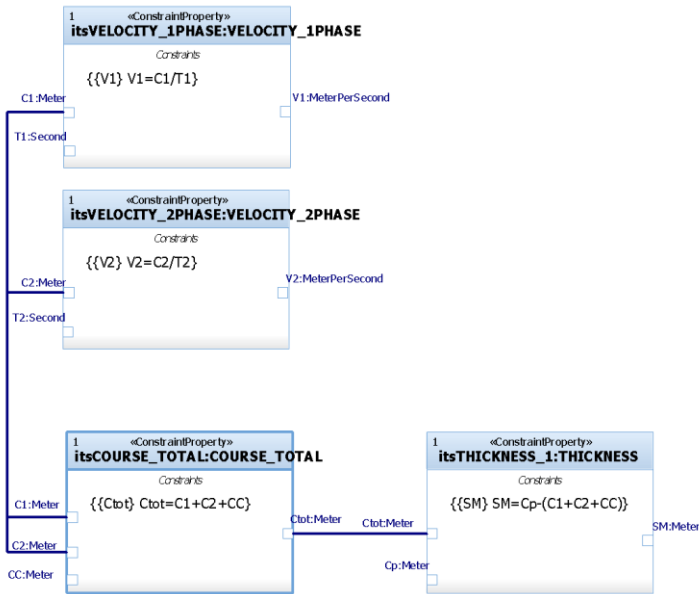


Figure 18: Invariant Modelling Patterns out of the RCA analysis

The physical meaning of each data-driven pattern has been defined using the models discussed in paragraph 4.2. Each pattern has been formalised according to the template presented in section 3.5 and represented in Table 7.

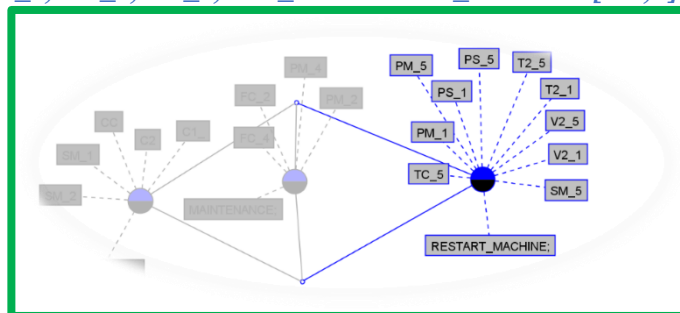
Table 7: The List of Invariant Modeling Patterns identified by RCA Data-driven Analyses based on SysML models

| P1: MOULD FILLING PATTERN | |
|--|--|
| <p>1. ID: P1</p> <p>2. PATTERN NAME: MOULD FILLING PATTERN</p> <p>3. DESCRIPTION: the pattern aims to show and describe the correlations between the parameters C1, C2, and CC. The pattern represents the course of the plunger for filling the mould. This impacts the quality of the product (SM).</p> | |
| <p>4. P1 EXAMPLE OF ASSOCIATION RULE (Concept):</p> <p>1: C1,C2,CC,SM_1 □ Microstop [0.5,1]</p> | |
|  | |
| <p>The set of data includes:</p> <ul style="list-style-type: none"> ● Plunger course C_1 (mm) in the first phase of the injection stage ● Plunger course C_2 (mm) in the second phase of the injection stage ● Plunger course CC (mm) in the multiplied phase of the injection stage ● Product thickness SM (mm²) | |
| <p>5. P1 SYML VIEW:</p> | <p style="text-align: right;">MODEL</p>  |
| <p>6. P1 APPLICABILITY: Use mould filling pattern for:</p> <ul style="list-style-type: none"> ● Modelling and preventing problems of micro-stop. ● Modelling and preventing problems during filling processes. ● Modelling the behaviour of the plunger course. ● Modelling the behaviour of an injection system. | |

P2: MACHINE RESTART PATTERN

1. ID: P2
2. PATTERN NAME: MACHINE RESTART PATTERN
3. DESCRIPTION: the pattern aims to show and describe the correlations between V_2 , T_2 , PM, PS, SM, TC. The pattern denotes the restart of the machine after a machine downtime. The restart of a machine generates a recasting of the product.

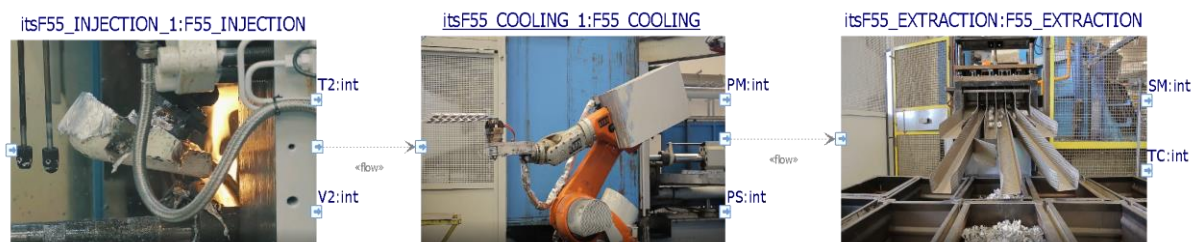
4. P2 EXAMPLE OF ASSOCIATION RULE (Concept):
 $TC_5, PM_1, PS_1, V2_5, T2_5, SM_5 \square Restart_Machine [0.5,1]$



The set of data includes:

- Plunger speed V_2 (m/sec) in the second phase of the injection stage.
- Plunger time T_2 (msec) in the second phase of the injection stage.
- Multiplied pressure PM (Pa) in the multiplied phase of the injection stage
- Specific pressure PS (Pa)
- Product thickness SM (mm²)
- Cycle time TC (sec)

5. P2 SYSML MODEL VIEW:



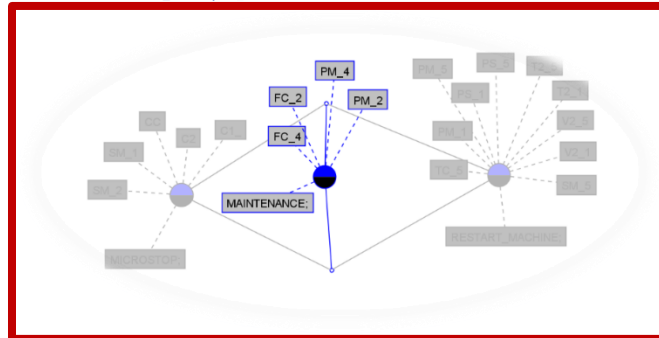
6. P2 APPLICABILITY: Use machine restart pattern for:
 - Modelling and preventing problems of recasting.
 - Modelling the restart of a machine.

P3 CLAMPING SYSTEM PATTERN

1. ID: P3
2. PATTERN NAME: CLAMPING SYSTEM PATTERN
3. DESCRIPTION: the pattern aims to show and describe the correlations between PM, PS, and FC. The pattern identifies the cooling stage of the die-casting aluminium process. The multiplied pressure and the clamping force are the parameters for monitoring the state of the clamping system. The clamping force FC and multiplied pressure PM values can generate mechanical problems.

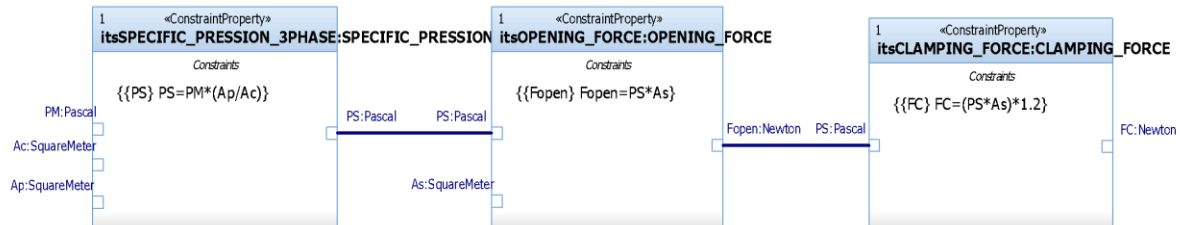
4. P3 ASSOCIATION RULE (Concept):

3: $PM_2, FC_2 \square Maintenance [0.5,1]$



The set of data includes:

- Clamping Force FC (kN)
- Multiplied Pressure PM (Pa)



5. P3 SYSML MODEL VIEW

6. P3 APPLICABILITY: Use clamping system pattern for:

- Modelling the behaviour of a clamping system (e.g., mould clamping).
- Modelling and preventing mechanical breakdown.

A Web platform based on SQL language in the Oracle database has been created, as shown in Figure 19. Structured Query Language (SQL) is the standard language that can be used for storing, manipulating and retrieving data stored in a relational database to 1) Execute queries against the database; 2) Create stored procedures in a database; 3) Create dashboards to represent the meaning of each data-driven construct. In this case, SQL has been applied for formalising and instantiating the patterns described in Table 7, creating, in this way, analytical reports for the employees.

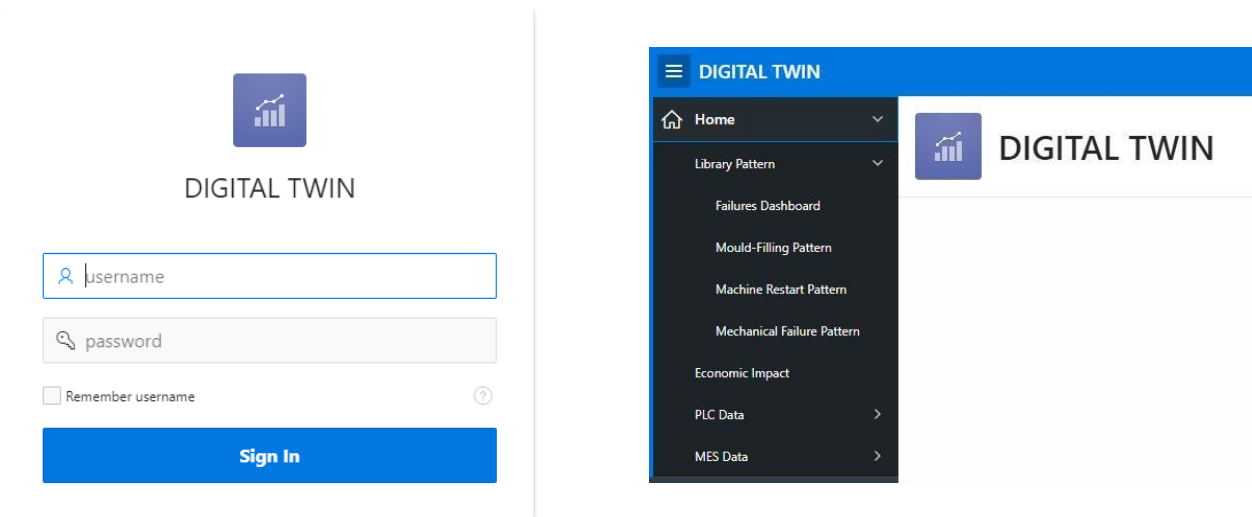


Figure 19: Web Platform for Data-driven patterns

4.6.1 MOULD FILLING PATTERN (P1)

The pattern P1 has been formalised in the web platform to describe and analyse the micro-stop machine downtime. The pattern presents the correlation between the following parameters:

- Plunger course C_1 (mm) in the first phase of the injection stage
- Plunger course C_2 (mm) in the second phase of the injection stage
- Plunger course CC (mm) in the multiplied phase of the injection stage
- Product thickness SM (mm²)

The daily micro-stop dashboard shown in Figure 20 allows an understanding of the frequency of this problem over time, the duration and the economic impact of each occurrence.

It is possible to click on each occurrence to understand the degradation trends of SM over time, as shown in Figure 21. The red line indicates the threshold value of SM (SM_{Min}). **When SM is equal to or below SM_{Min} , the injection cycle (≈ 36 products) is recast.** The dashboard shows that a recasting always precedes a micro-stop. The requirement is to predict SM before the threshold value. It is necessary to discover the relationship between SM , C_1 , C_2 and CC . In particular, the dashboard Pattern Values in Figure 22 demonstrate that C_1 and C_2 are inversely proportional as a function of SM . All possible combinations between C_1 , C_2 , CC and SM need to be explored to define how to detect all possible behaviour of the system.

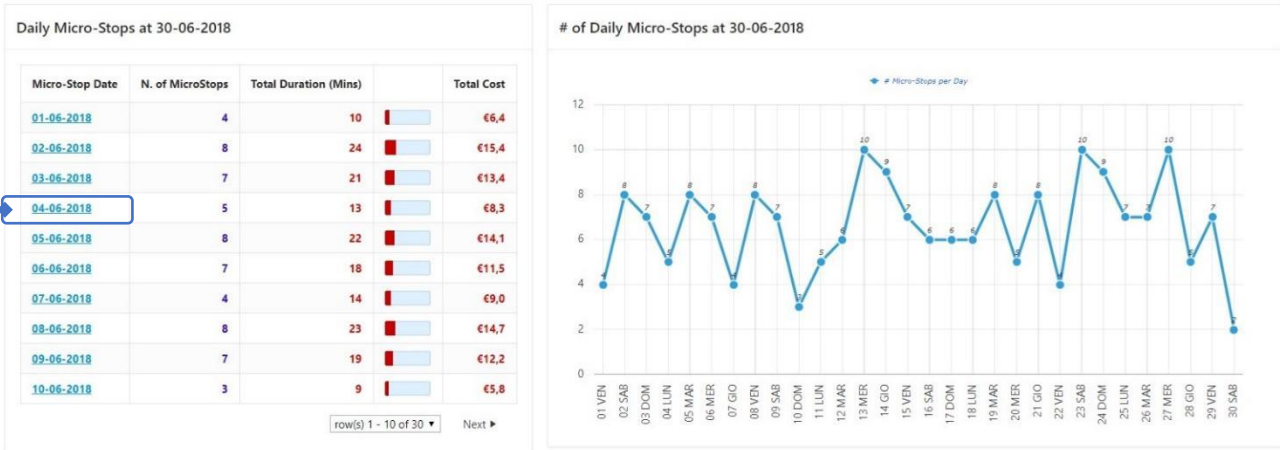


Figure 20: Daily Micro-stop

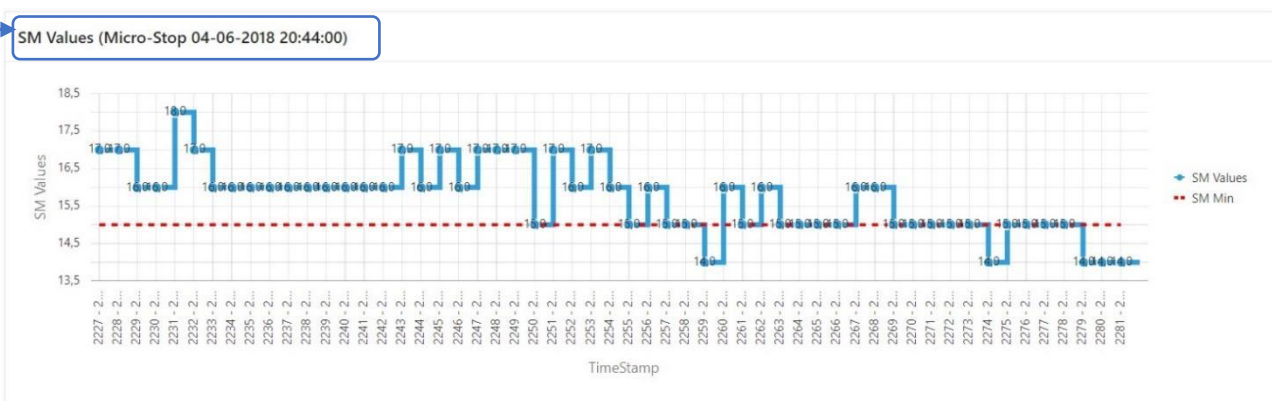


Figure 21: SM Values



Figure 22: Pattern Values

All correlations are split into three main categories, as shown in Figure 23:

- The green area indicates all possible combinations between C1, C2 and CC when SM is greater than SM_Min. These combinations define when the process is functioning correctly.
- The yellow area indicates all possible combinations between C1, C2 and CC when SM is close to the threshold value.
- The red area indicates all possible combinations between values C1, C2 and CC when SM is equal to or below SM_Min. These combinations indicate a system failure and the machine's consequent micro-stop.

For the case study under analysis, the pattern P1 can be instantiated to predict possible micro stop downtimes avoiding quality problems and product recasting.

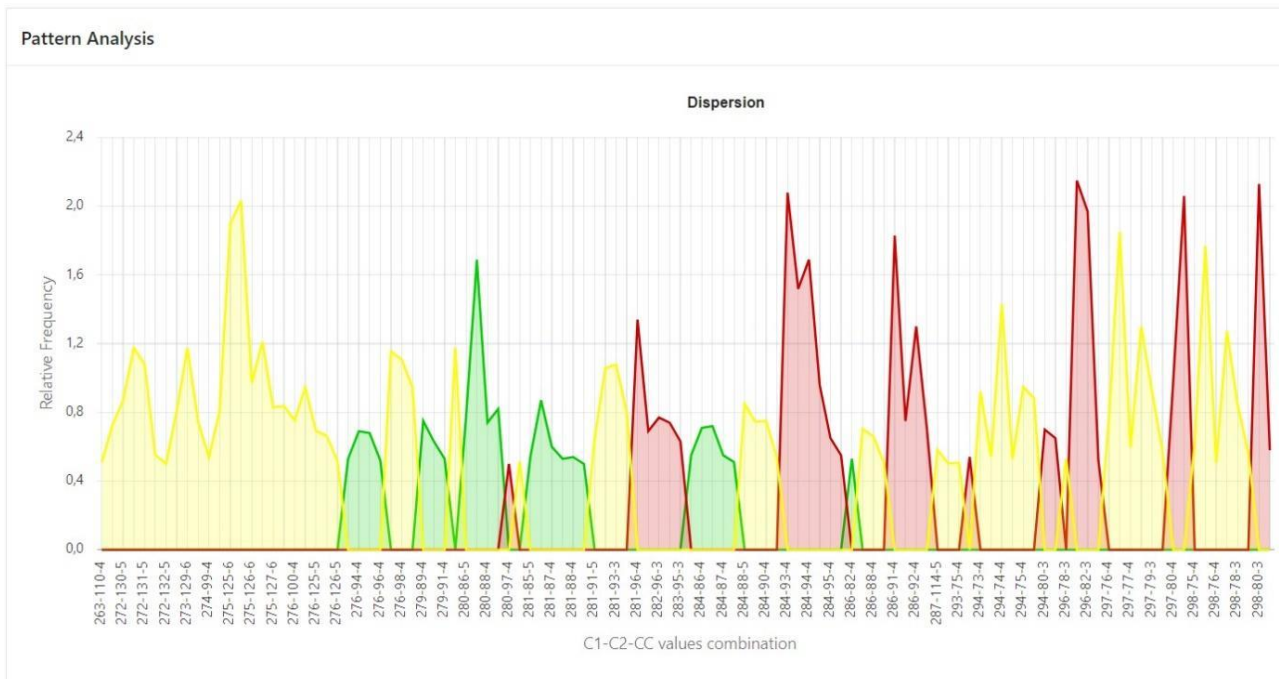


Figure 23: Pattern Analysis for Prediction

4.6.2 MACHINE RESTART PATTERN (P2)

The pattern P2 has been designed in the platform to explore the restart behaviour of the die-casting process. The pattern presents the correlation between the following parameters:

- Plunger speed V_2 (m/sec) in the second phase of the injection stage.
- Plunger time T_2 (msec) in the second phase of the injection stage.
- Multiplied pressure PM (Pa) in the multiplied phase of the injection stage
- Specific pressure PS (Pa)
- Product thickness SM (mm²)
- Cycle time TC (sec)

The dashboard in Figure 24 has been created to demonstrate that whenever a downtime occurs, the restart of the machine presents one or more parameters above listed, out of the threshold value. The parameters V_2 , T_2 PM and PS could present values greater than or below the threshold value, while SM and TC only present values greater than a threshold value. It is possible to click on each downtime problem (cause code), as shown in Figure 25, to analyse which parameter causes it frequently and which value assumes. In this case, the instantiation of the pattern P2 helps prevent downtimes, avoiding possible product recasting caused by machine restarting.

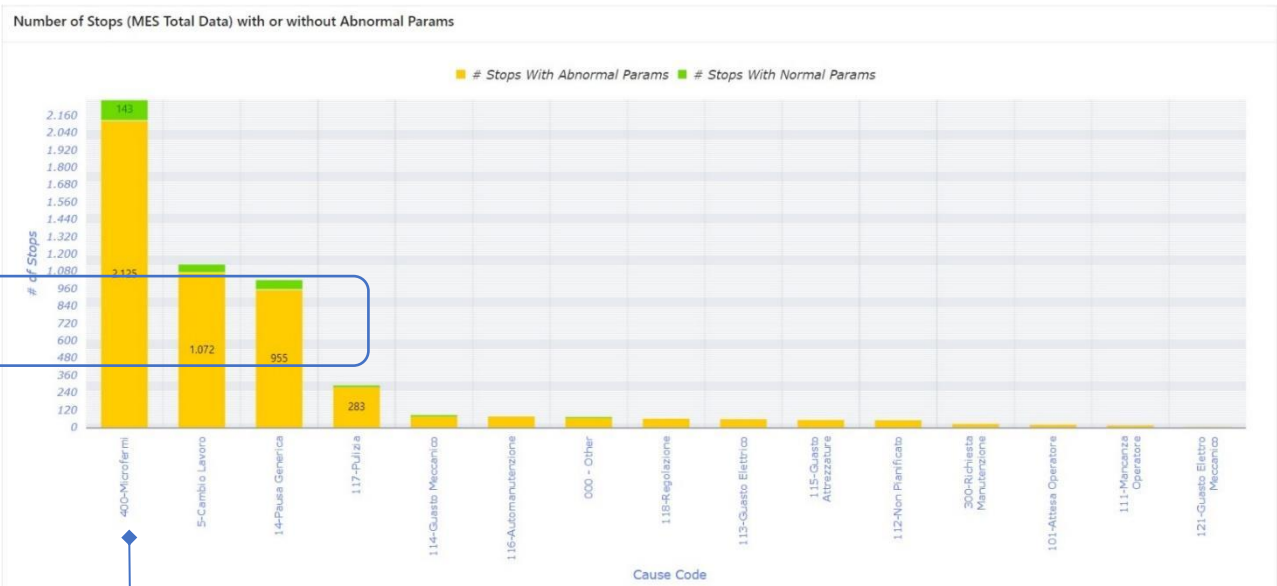


Figure 24: Number of Stops Machine

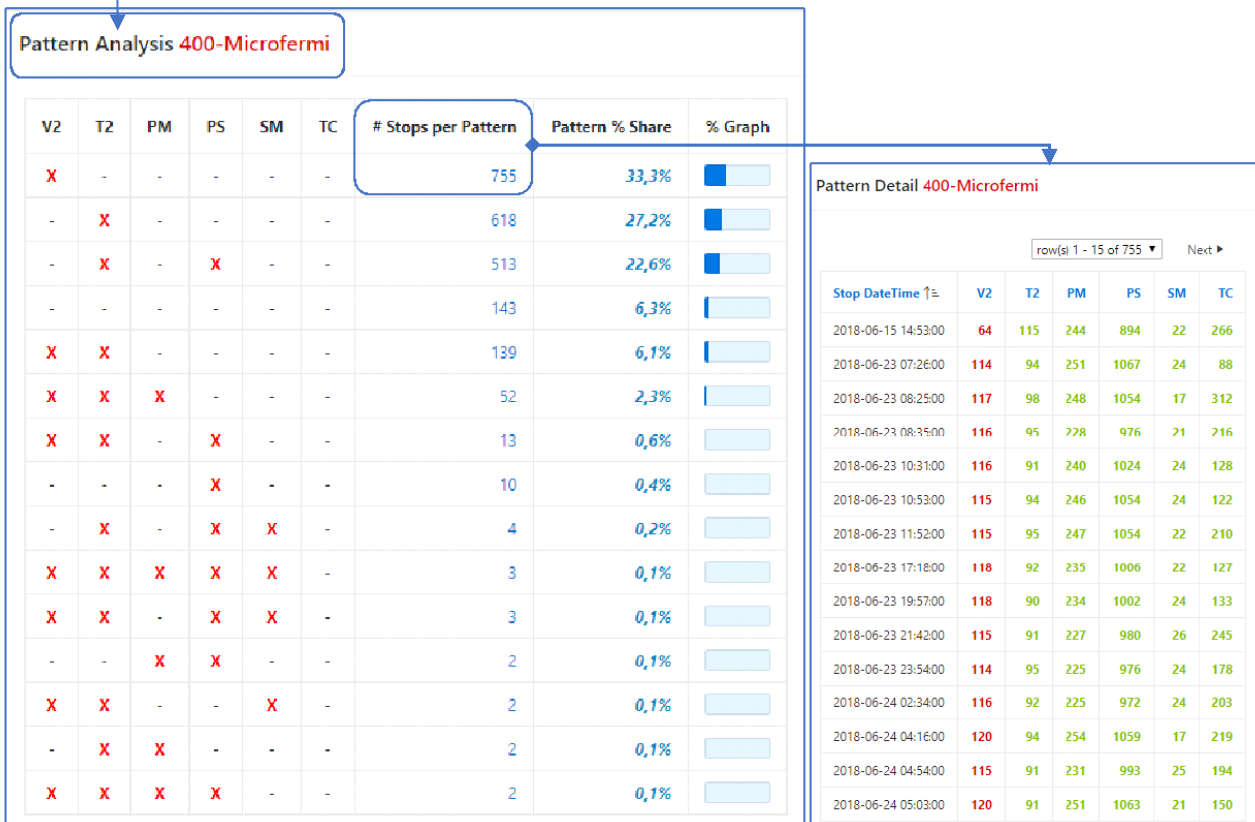


Figure 25: Restart Machine Pattern Analysis

4.6.3 CLAMPING SYSTEM PATTERN (P3)

The pattern P3 has also been designed in the platform to evaluate the correlation between the following two parameters:

- Clamping Force FC (kN)
- Multiplied Pressure PM (Pa)

The dashboard percentage of mechanical failure in Figure 26 has been created to comprehend the number of occurrences of mechanical maintenance for each month.

The yellow dashed lines represent the thresholds values. Each intended line represents the trend of each mechanical maintenance downtime occurrence over time. The result is that FC is inversely proportional to PM. It means that if an occurrence presents values of FC or PM above or below the line, it is possible to detect possible incoming maintenance problems. In the first scenario in Figure 27, the first four occurrences of June 2018 have been investigated. PM operate below the lower range than FC is in range. In the second scenario in Figure 28, the last four June 2018 have been investigated to demonstrate that the situation is inverse in this case. FC operate above the upper range than PM is in the range.

The instantiation of this last pattern (P3) can help predict failures of the process caused by mechanical maintenance.

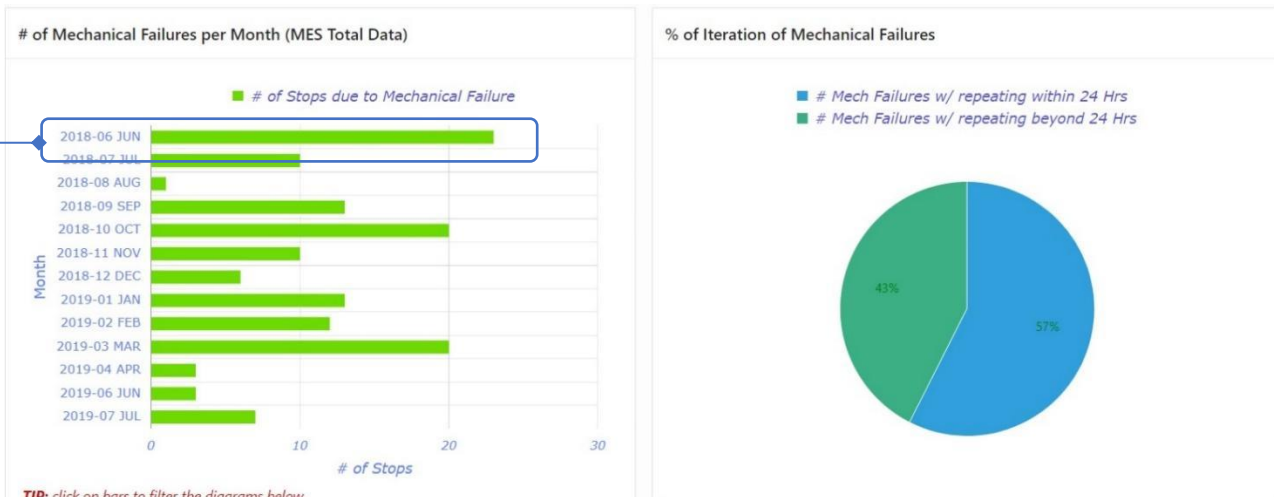


Figure 26: Mechanical Failures

The first scenario



Figure 27: Scenario I: FC and PM Trends over Time

The second scenario

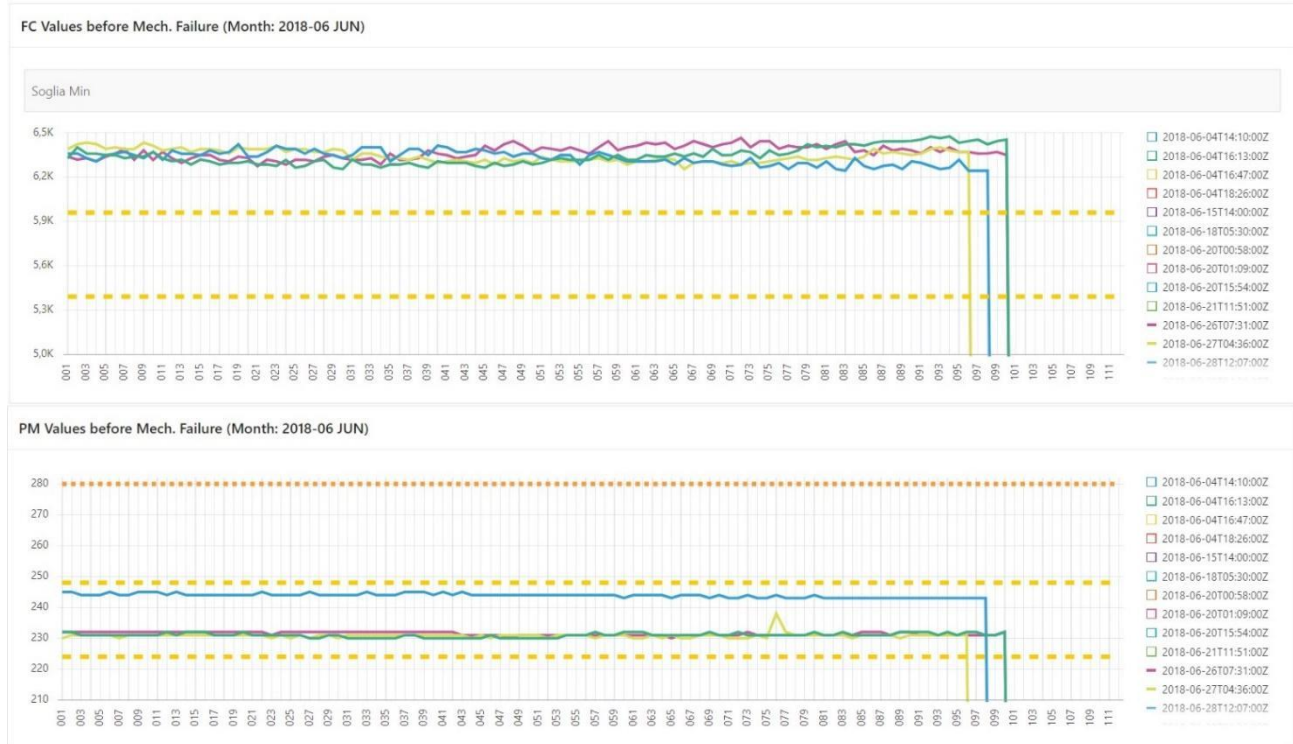


Figure 28: Scenario II: FC and PM Trends over Time

4.7 CRITERIA FOR DIGITAL TWIN DESIGN

The patterns described (P1, P2, P3) in sections 4.5 and 4.6 have been instantiated to design the digital twin of the die-casting process to predict micro-stop and mechanical problems, as shown in Figure 29. The proposed solution consists of a realistic production environment but is “augmented” with intrinsic technological knowledge. With our approach, the physical settings interact with the digital space, according to specific properties and rules, to understand:

- The behaviour of the process.
- The correlations between technological parameters.
- The correlations between parameters and effects like quality defects, maintenance problems.

The digital twin has been designed to support the employees in the decision-making process to:

- Identify the several quality problems of the components autonomously, compared to the standards (dimensions, tolerances, finishes, quantity).
- Alert operators through proper alarm systems about abnormal or out-of-tolerance situations.
- Analyse and correlate the symptoms and causes of failures and defects in production.
- Support the choice of corrective actions to eliminate the detected failures and defects.

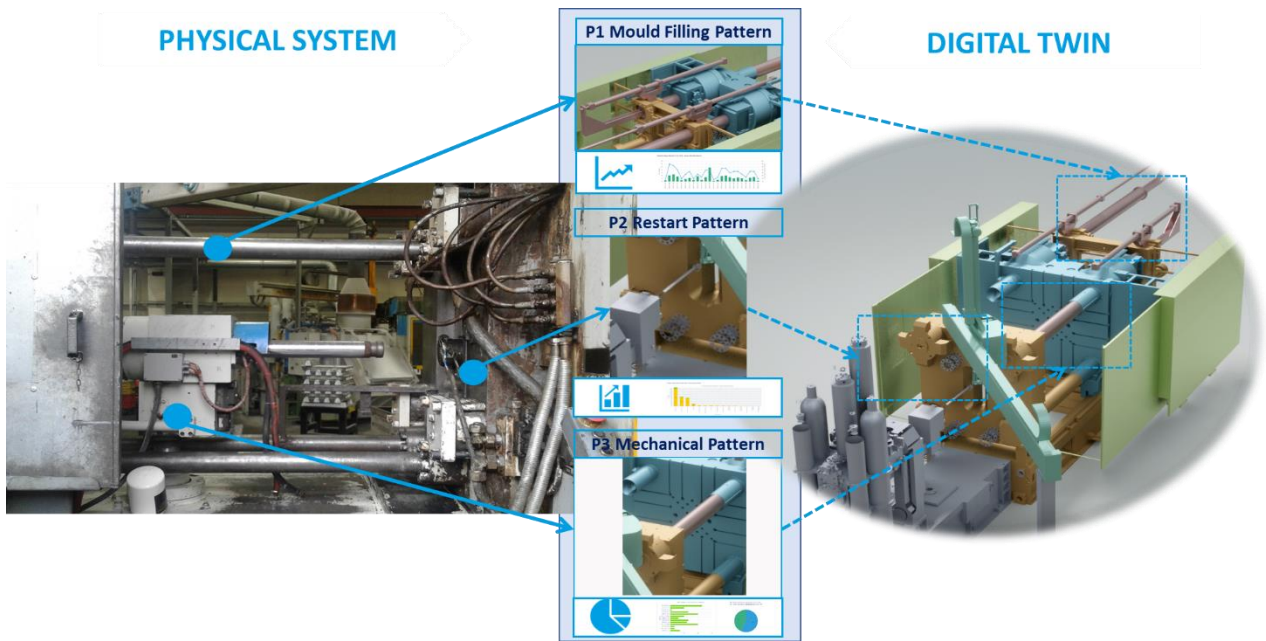


Figure 29: Digital Twin of Die-casting Aluminium Process

The gathered data from the shop floor level are aggregated, analysed and interpreted in the digital environment according to the conditions listed in Table 8. The conditions are defined and formalised based on the platform's patterns shown and analysed. A condition is associated with a possible state that a digital twin can detect and recognise.

Table 8: DT States based on the P1, P2, and P3 Patterns

| STATE_ID | STATE_NAME | STATE_CONDITIONS | DECISION-SUPPORT (ACTION TO DO) | ID PATTERN |
|----------|----------------------------------|--|--|----------------------------|
| <u>0</u> | "Machine Works without Problems" | = | If the DT detects STATE 0, it will understand that process is working properly. | = |
| <u>1</u> | "Prediction Micro-Stop" | $SM_LimLow \leq SM \leq SM_LimUp$ _p | If the DT detects STATE 1, it will suggest that the employees should clean the machine. If the action will be not performed there will be a recasting in 5minutes. | Mould Filling Pattern (P1) |

| | | | | |
|----------|---|--|---|------------------------------|
| <u>2</u> | "Recasting for Micro-Stop" | V2<=LimLow OR V2>=LimUpp OR T2<=LimLow OR T2>= LimUpp PM<=LimLow OR PM>=LimUpp OR PS<=LimLow OR PS >=LimUpp SM>=LimUpp OR TC>=LimUpp | If the DT detects STATE 2, it will understand that there is a Recasting due to a Restart Machine | Machine Restart Pattern (P2) |
| <u>3</u> | "Prediction Mechanical Maintenance" | LimLow<=PM<=LimUpp OR LimLow<=PM<=LimUpp OR LimLow<=FC<=LimUpp OR LimLow<=FC<=LimUpp | If the DT detects STATE 3, it will suggest that the employees should perform mechanical maintenance on the mould otherwise, a machine downtime will occur in one hour (estimated time of intervention 1.5h) | Clamping System Pattern (P3) |
| <u>4</u> | "Recasting for Micro-Stop" | SM<=LimLow | If the DT detects STATE 4, it will understand that there is a Recasting due to a micro-stop | Mould Filling Pattern (P1) |
| <u>5</u> | "Stop Machine for Mechanical Maintenance" | FC<=LimLow OR FC>=LimUpp | If the DT detects STATE 5, it will understand that the machine is on downtime due to maintenance. | Clamping System Pattern (P3) |

The knowledge extracted from patterns has been encapsulated in the designed algorithm below that supports the digital twin in judging, evaluating, optimising, and/or predicting all possible states based on predefined conditions explored in sections 4.6.1-4.6.2.

DT Algorithm built on P1,P2 and P3 STATES AND CONDITIONS:

→Verify State "Prediction Micro-Stop"

IF cur_stringa.SM BETWEEN LimLow AND LimUpp THEN v_stato_rkd := 1; END IF;

→Verify State "Re-start Machine"

IF v_stato_rkd = 0 AND

(cur_stringa.V2 <= LimLow OR cur_stringa.V2 >= LimUpp

OR cur_stringa.T2 <= LimLow OR cur_stringa.T2 >= LimUpp

OR cur_stringa.PM <= LimLow OR cur_stringa.PM >= LimUpp

OR cur_stringa.PS <= LimLow OR cur_stringa.PS >= LimUpp

OR cur_stringa.SM >= LimUpp

```

OR cur_stringa.TC >= LimUpp
)
THEN v_stato_rkd := 2;
END IF;

```

→Verify State "Prediction Mechanical Maintenance"

```

IF v_stato_rkd = 0 AND cur_stringa.TIMESTAMPLOCAL >= TO_DATE('01-03-2022','DD-MM-YYYY')
AND
(cur_stringa.PM BETWEEN LimLow AND LimUpp)
OR
(cur_stringa.FC BETWEEN LimLow AND LimUpp) OR (cur_stringa.FC BETWEEN LimLow AND
LimUpp)
)
THEN
v_stato_rkd := 3;
END IF;
IF v_stato_rkd = 0 AND cur_stringa.TIMESTAMPLOCAL < TO_DATE('01-03-2022','DD-MM-YYYY') AND
( (cur_stringa.PM LimLow AND LimUpp) OR
(cur_stringa.FC BETWEEN LimLow AND LimUpp) OR (cur_stringa.FC BETWEEN LimLow AND
LimUpp)
)
THEN

```

Contribution to the Formalisation of Data-Driven Invariant Modelling Patterns of Cyber-Physical Systems
106

```

v_stato_rkd := 3;
END IF;

```

→Verify "Recasting for Micro-Stop"

```

IF v_stato_rkd = 0 AND
( cur_stringa.SM <= LimLow
)
THEN
v_stato_rkd := 4;
END IF;

```

→Verify State "Stop Machine for Mechanical Maintenance"

```

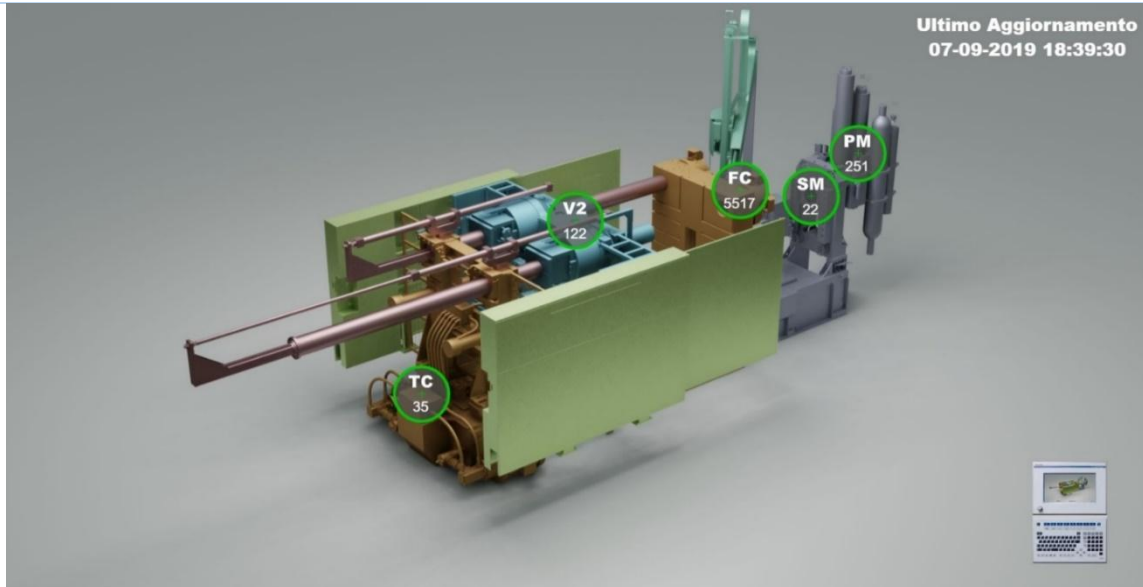
IF v_stato_rkd = 0 AND
( cur_stringa.FC <= LimLow OR cur_stringa.FC >= LimUpp
)
THEN
v_stato_rkd := 5;
END IF;

```

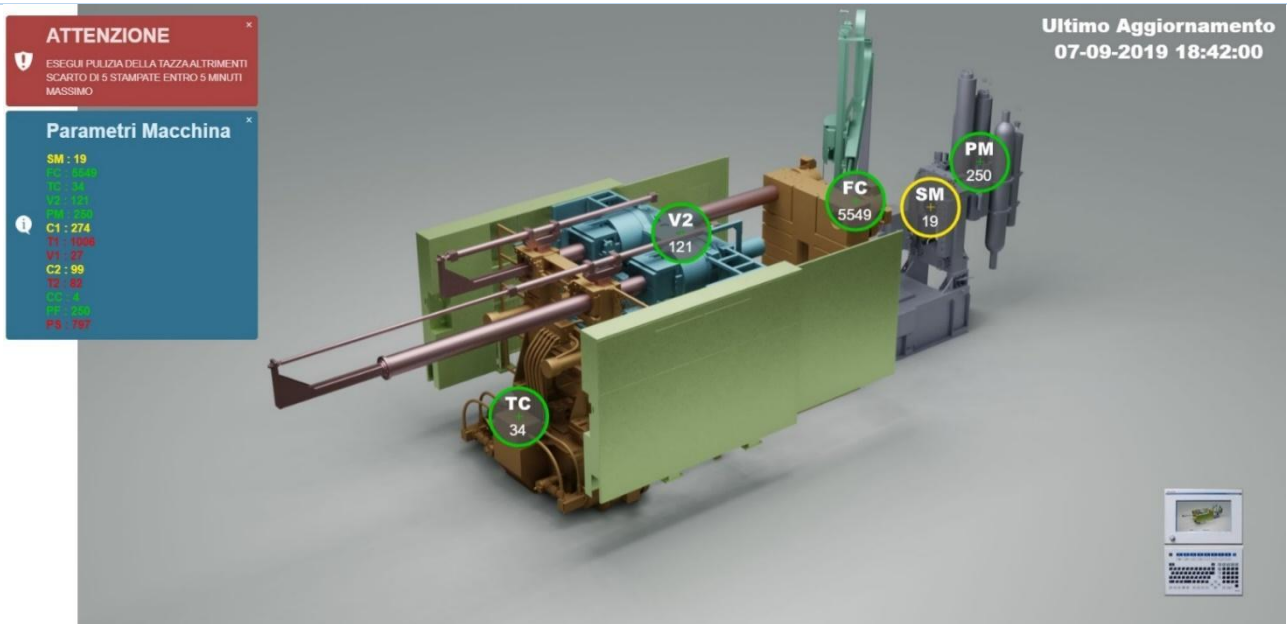
The digital twin can receive, read and evaluate possible performances (states) of the process based on the values of the operating parameters. The digital twin can recognise and predict four different behaviours (performance) of the die-casting process in the following sequence described in Figure 30:

DIGITAL TWIN OF DIE CASTING ALUMINIUM

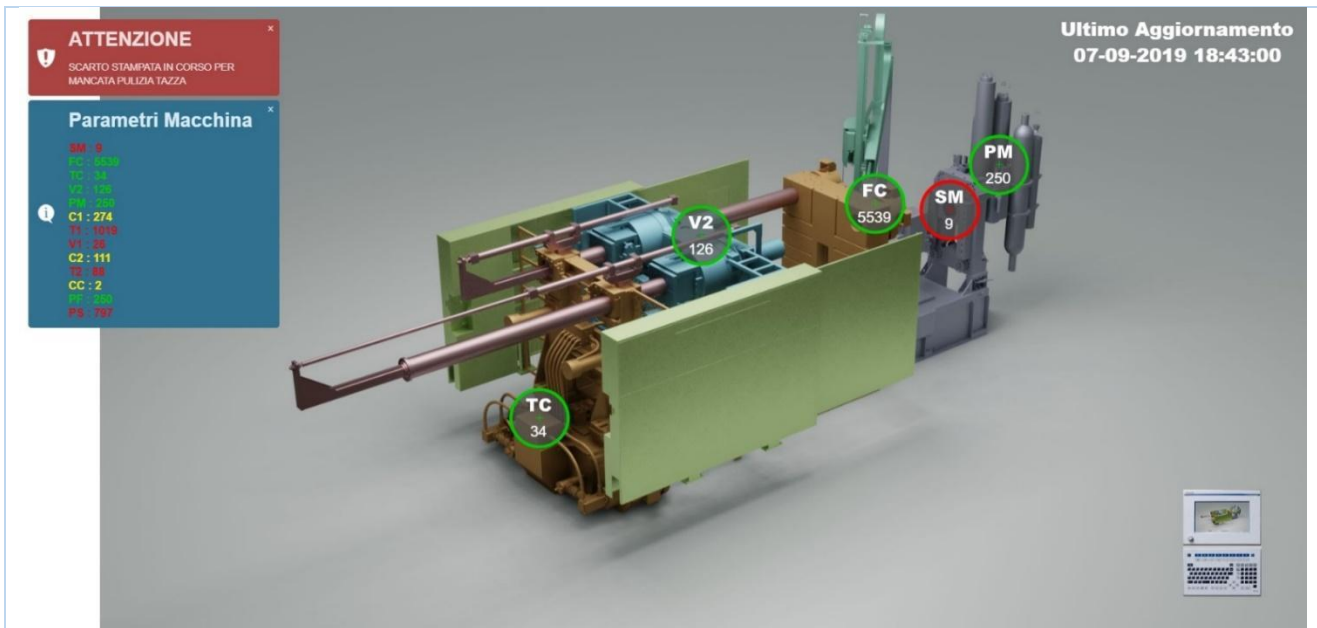
1. At 07-09-2019 18:39:30 The Digital Twin does not detect any problem (RECOGNIZE STATE 0)



2. At 07-09-2019 18:42:00 The digital twin predicts that a micro-stop problem will happen in 5 minutes based on C1, C2, CC and SM values. The DT prescribes to the employees which action performs. (RECOGNIZE STATE 1)



3. At 07-09-2019 18:43:00: The digital twin informs the employees that a problem of micro-stop occurred because they did not perform the action prescribed in the state 1. (RECOGNIZE STATE 4)



4. At 07-09-2019 18:46:00 The digital twin detects the restart machine after 3 minutes of downtime due to the value of V2 and TC.
(RECOGNISE STATE 2)

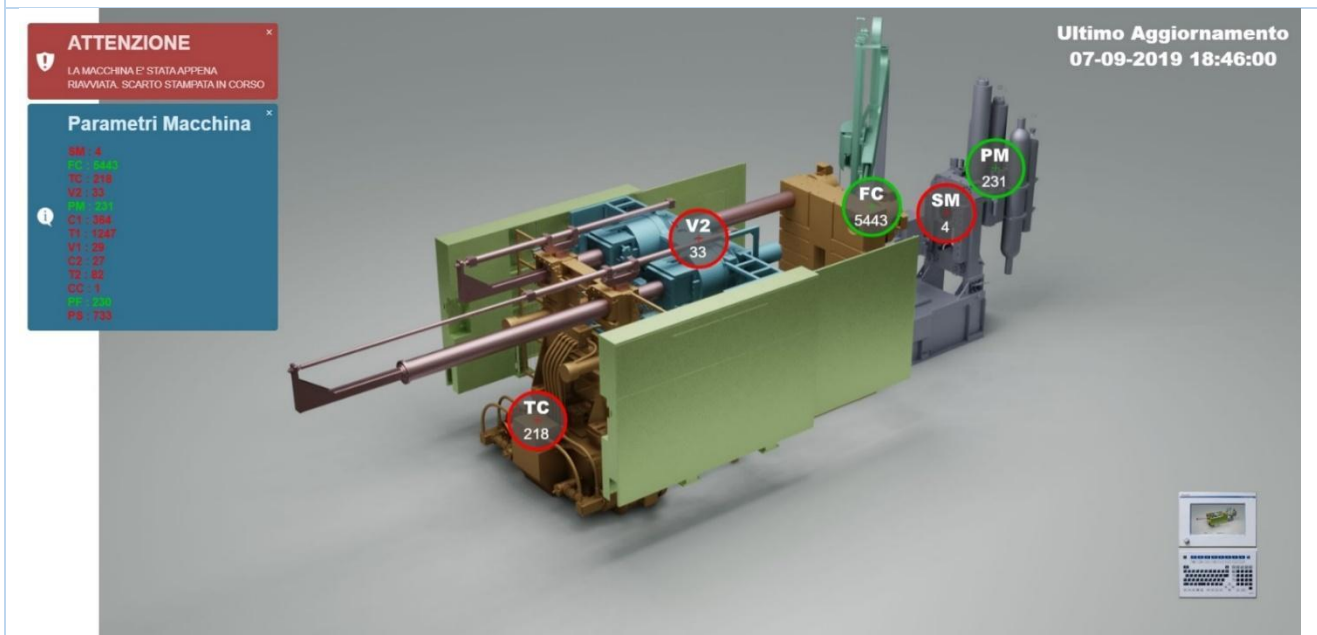


Figure 30: Digital Twin of the die-casting Aluminium Process based on Patterns P1,P2 and P3

4.8 Knowledge Re-use

The formalized knowledge can be used to design the digital twin of other manufacturing processes based on the knowledge formalized in Table 7.

5. DISCUSSION AND CONCLUSIONS

According to the literature review performed in section 1, one significant limitation is that modelling a digital copy of physical systems is quite complex, and generally, the modelling action has a specific

validity depending on the application type. The core DT-modelling methodology presented in this paper is to create invariant modelling patterns extracted from a data-driven analysis to have the chance to re-use them as macro-programs in different applications. This may bring an enormous value of standardisation over different technological applications and a deeper understanding of related problems. The paper presents a new methodology approach for discovering data-driven modelling patterns for digital twin design. The formalisation approach presented in section 3, for discovering data-driven patterns goes in this direction. It involves the combination of data-driven and model-based approaches with design patterns to define and identify invariant modelling. The iterative approach consists of eight different stages appropriately coordinated to make this almost usable in any production case.

The case study presented to demonstrate the validity of the methodology was firstly defined in SysML to describe the requirements, the structure, and the related behaviour. Adopting the relation concept analysis (RCA) method was the key to automatically detecting tacit associations in data and thus, automatically detecting invariant modelling patterns.

A digital twin prototype has been developed as in section 4 using the proposed methodology presented in section 3 and instantiating the data-driven patterns discussed in sections 4.4 and 4.5.

Pattern detection can be a prerequisite for building an intelligent system like a Digital Twin. The DT consists of a set of accurate models with complex structures and behaviour that reflect the real-time operations of the physical system to predict possible failures. The literature review does not present a uniform modelling approach for DT's. Even though multiple tools and methods have been proposed, a consistent modelling procedure is still missing, which can support the demand-oriented selection of tools/methods for DT modelling based on established DT modelling requirements. It is difficult to build an accurate model for a DT using model-based approaches due to the complexity of a physical system. The lack of a univocal reference architecture leads to developing Digital Twin solutions using different approaches, models and data.

Standard Digital Twin solutions would be desirable to provide design criteria and constraints with reference architectural aspects, reference information model and communication protocols are clearly defined. The proposed DT-design methodology allows deriving a criterion to self-detect modelling patterns that can be used (and re-used) to create digital models of different systems or processes. The approach combines a model-based with a data-driven technique. SysML (model-based) that is part of our methodology allowed to model and formalize a manufacturing system's static structure and dynamic behaviour. RCA (data-driven) also adopted served to detect invariant modelling patterns, which can be applied to any manufacturing system. Our approach relies on the invariance recognition in data in the form of modelling patterns. Recognising the invariance allows the DT modelling efficiency, flexibility and reusability in several different manufacturing systems. A manufacturing system here is in fact conceived as a set of manufacturing operations where the inputs (X_i) are the operating parameters, and the outputs (Y_i) are the performance. This generalization allows the patterns detection that can be instantiated for designing and modelling a DT: a pattern here discovers a possible configuration of operating parameters that generate a specific performance.

In the case study discussed, the discovered patterns defined which technological parameters combination causes a potential downtime, where this latter represents a degradation of performance. This technological knowledge, once instantiated, defines how to build a DT able to recognize a potential system degradation, and predict and prevent this latter based on the operating profile.

Future research may be devoted to enrich the pattern's semantics and to create a comprehensive library of formalised data-driven patterns. Other data sets, such as logistic, product, and customer data, need to be selected to extract new data-driven constructs and create a consistent library of patterns.

Acknowledgement

The Italian Ministry of Education supported this work, University and Research under the Programme "Department of Excellence" Legge 232/2016 (Grant No. CUP - D94I18000260001).

The authors are grateful to the Italian Company Master Italy s.r.l for partially funding this research.

CRedit authorship contribution statement

Concetta SEMERARO: Investigation, Formal analysis, Writing–original draft, Methodology, Developing Prototype

Michele DASSISTI: Conceptualization, Funding acquisition, Supervision, Writing–review&editing.

Hervé PANETTO: Methodology, Supervision, Rewrite and Edit

Mario LEZOCHÉ: Methodology, Writing

Author CS was responsible for the study conduction and assimilation of the literature to select the final sample, defining the methodological approach and creating the DT prototype for the case study;

Authors Michele DASSISTI and Concetta Semeraro are completely responsible for conceptualising the idea within and developing paragraphs 2 and 4.

Authors ML, HP, MD assessed the quality of the included studies; CS wrote a first draft of the manuscript; ML, HP, MD contributed to the final version and provided several suggestions to improve the quality of the research paper. All authors have read and agreed to the paper being submitted in the present form.

Declaration of Competing Interest

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

References

- [1] M. P. Uysal e A. E. Mergen, «Smart manufacturing in intelligent digital mesh: Integration of enterprise architecture and software product line engineering», *Journal of Industrial Information Integration*, vol. 22, pag. 100202, giu. 2021, doi: 10.1016/j.jii.2021.100202.
- [2] J. A. Estefan, «Survey of model-based systems engineering (MBSE) methodologies», *IncoSE MBSE Focus Group*, vol. 25, n. 8, pagg. 1–12, 2007.
- [3] C. Semeraro, M. Lezoche, H. Panetto, e M. Dassisti, «Digital Twin Paradigm: A Systematic Literature Review», *Computers in Industry*, vol. 130, pag. 103469, 2021.
- [4] T. Ruppert e J. Abonyi, «Integration of real-time locating systems into digital twins», *Journal of Industrial Information Integration*, vol. 20, pag. 100174, dic. 2020, doi: 10.1016/j.jii.2020.100174.
- [5] A. K. Ghosh, A. S. Ullah, R. Teti, e A. Kubo, «Developing sensor signal-based digital twins for intelligent machine tools», *Journal of Industrial Information Integration*, vol. 24, pag. 100242, dic. 2021, doi: 10.1016/j.jii.2021.100242.
- [6] J. Lee, B. Bagheri, e H.-A. Kao, «A cyber-physical systems architecture for industry 4.0-based manufacturing systems», *Manufacturing Letters*, vol. 3, pagg. 18–23, 2015.

- [7] C. Semeraro, M. Lezoche, H. Panetto, e M. Dassisti, «Pattern-based Digital Twin for Optimizing Manufacturing Systems: A Real Industrial-case Application», in *17th IFAC Symposium on Information Control Problems in Manufacturing*, 2021.
- [8] C. Semeraro, M. Lezoche, H. Panetto, M. Dassisti, e S. Cafagna, «Data-driven pattern-based constructs definition for the digital transformation modelling of collaborative networked manufacturing enterprises», in *Working Conference on Virtual Enterprises*, 2019, pagg. 507–515.
- [9] L. F. Konikow e J. D. Bredehoeft, «Ground-water models cannot be validated», *Advances in water resources*, vol. 15, n. 1, pagg. 75–83, 1992.
- [10] V. Venkatasubramanian, R. Rengaswamy, e S. N. Kavuri, «A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies», *Computers & chemical engineering*, vol. 27, n. 3, pagg. 313–326, 2003.
- [11] J. W. Schmidt, *Simulation and analysis of industrial systems*. Homewood (Ill.), Etats-Unis d’Amérique: R. D. Irwin, 1970.
- [12] D. A. Tobon-Mejia, K. Medjaher, N. Zerhouni, e G. Tripot, «A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models», *IEEE Transactions on reliability*, vol. 61, n. 2, pagg. 491–503, 2012.
- [13] J. Luo, M. Namburu, K. Pattipati, L. Qiao, M. Kawamoto, e S. Chigusa, «Model-based prognostic techniques [maintenance applications]», in *Proceedings AUTOTESTCON 2003. IEEE Systems Readiness Technology Conference.*, 2003, pagg. 330–340.
- [14] M. Pecht, «Prognostics and health management of electronics», *Encyclopedia of structural health monitoring*, 2009.
- [15] P. Nowack e M. E. Caspersen, «Model-based thinking and practice: A top-down approach to computational thinking», in *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, 2014, pagg. 147–151.
- [16] International Council on Systems Engineering (INCOSE), «International Council on Systems Engineering (INCOSE), Systems Engineering Vision 2020, Version 2.03, TP-2004-004-02, September 2007.», 2007. <http://www.omg.org/spec/OCL/>
- [17] ISO/IEC 19514, «ISO/IEC 19514:2017», *ISO*, 2017. <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/52/65231.html> (consultato 6 marzo 2020).
- [18] A. M. Madni, C. C. Madni, e S. D. Lucero, «Leveraging digital twin technology in model-based systems engineering», *Systems*, vol. 7, n. 1, pag. 7, 2019.
- [19] G. Bachelor, E. Brusa, D. Ferretto, e A. Mitschke, «Model-Based Design of Complex Aeronautical Systems Through Digital Twin and Thread Concepts», *IEEE Systems Journal*, 2019.
- [20] S. Boschert, C. Heinrich, e R. Rosen, «Next generation digital twin», in *Proc. TMCE*, 2018, pagg. 209–217.
- [21] A. M. Madni, C. C. Madni, e S. D. Lucero, «Leveraging digital twin technology in model-based systems engineering», *Systems*, vol. 7, n. 1, pag. 7, 2019.
- [22] M. Schluse, L. Atorf, e J. Rossmann, «Experimentable digital twins for model-based systems engineering and simulation-based development», in *Systems Conference (SysCon), 2017 Annual IEEE International*, 2017, pagg. 1–8.
- [23] J. Miller e J. Mukerji, «J. Miller and J. Mukerji. MDA Guide Version 1.0.1. Technical report, Object Management Group (OMG),2003.», 2003. https://www.omg.org/news/meetings/workshops/UML_2003_Manual/00-2_MDA_Guide_v1.0.1.pdf
- [24] L. Zhang, «Aspect-oriented development method for non-functional characteristics of cyber physical systems based on MDA approach.», *JSW*, vol. 7, n. 3, pagg. 608–619, 2012.
- [25] S. Chandhoke, T. Hayles, J. Kodosky, e G. Wang, «A model-based methodology of programming cyber-physical systems», in *2011 7th International Wireless Communications and Mobile Computing Conference*, 2011, pagg. 1654–1659.
- [26] J. Luo, K. R. Pattipati, L. Qiao, e S. Chigusa, «Model-based prognostic techniques applied to a suspension system», *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, n. 5, pagg. 1156–1168, 2008.
- [27] S. Friedenthal, A. Moore, e R. Steiner, *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann, 2014.
- [28] J. J. Gibson e L. Carmichael, *The senses considered as perceptual systems*, vol. 2. Houghton Mifflin Boston, 1966.
- [29] R. Zhang, V. Jayawardene, M. Indulska, S. Sadiq, e X. Zhou, «A data driven approach for discovering data quality requirements», 2014.
- [30] Y. Zhang, S. Ren, Y. Liu, T. Sakao, e D. Huisingh, «A framework for Big Data driven product lifecycle management», *Journal of Cleaner Production*, vol. 159, pagg. 229–240, 2017.
- [31] S. Jain, G. Shao, e S.-J. Shin, «Manufacturing data analytics using a virtual factory representation», *International Journal of Production Research*, vol. 55, n. 18, pagg. 5450–5464, 2017.
- [32] H. Al-Sahaf *et al.*, «A survey on evolutionary machine learning», *Journal of the Royal Society of New Zealand*, vol. 0, n. 0, pagg. 1–24, mag. 2019, doi: 10.1080/03036758.2019.1609052.

- [33] J. Zhang, S. O. Williams, e H. Wang, «Intelligent computing system based on pattern recognition and data mining algorithms», *Sustainable Computing: Informatics and Systems*, vol. 20, pagg. 192–202, dic. 2018, doi: 10.1016/j.suscom.2017.10.010.
- [34] D. P. Solomatine e A. Ostfeld, «Data-driven modelling: some past experiences and new approaches», *Journal of Hydroinformatics*, vol. 10, n. 1, pagg. 3–22, gen. 2008, doi: 10.2166/hydro.2008.015.
- [35] Y. Liu, T. Wang, H. Zhang, e V. Cheutet, «Simulation-based fuzzy-rough nearest neighbour fault classification and prediction for aircraft maintenance», *Journal of Simulation*, pagg. 1–15, 2019.
- [36] Y. Zhao, C. Zhang, Y. Zhang, Z. Wang, e J. Li, «A review of data mining technologies in building energy systems: Load prediction, pattern identification, fault detection and diagnosis», *Energy and Built Environment*, nov. 2019, doi: 10.1016/j.enbenv.2019.11.003.
- [37] P. Hansen e B. Jaumard, «Cluster analysis and mathematical programming», *Mathematical programming*, vol. 79, n. 1–3, pagg. 191–215, 1997.
- [38] R. Agrawal, T. Imieliński, e A. Swami, «Mining association rules between sets of items in large databases», in *Acm sigmod record*, 1993, vol. 22, pagg. 207–216.
- [39] M.-Y. Lin e S.-Y. Lee, «Improving the efficiency of interactive sequential pattern mining by incremental pattern discovery», in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, 2003, pagg. 8-pp.
- [40] Z. Jiang, Y. Guo, e Z. Wang, «Digital twin to improve the virtual-real integration of industrial IoT», *Journal of Industrial Information Integration*, vol. 22, pag. 100196, giu. 2021, doi: 10.1016/j.jii.2020.100196.
- [41] K. Tidiri, N. Chatti, S. Verron, e T. Tiplica, «Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges», *Annual Reviews in Control*, vol. 42, pagg. 63–81, 2016.
- [42] A. Kusiak, «Smart manufacturing», *International Journal of Production Research*, vol. 56, n. 1–2, pagg. 508–517, 2018.
- [43] E. B. Priyanka, S. Thangavel, X.-Z. Gao, e N. S. Sivakumar, «Digital twin for oil pipeline risk estimation using prognostic and machine learning techniques», *Journal of Industrial Information Integration*, pag. 100272, ago. 2021, doi: 10.1016/j.jii.2021.100272.
- [44] N. Sheibat-Othman, N. Laouti, J.-P. Valour, e S. Othman, «Support vector machines combined to observers for fault diagnosis in chemical reactors», *The Canadian Journal of Chemical Engineering*, vol. 92, n. 4, pagg. 685–695, 2014.
- [45] S. Frank *et al.*, «Hybrid model-based and data-driven fault detection and diagnostics for commercial buildings», National Renewable Energy Lab.(NREL), Golden, CO (United States), 2016.
- [46] H. Khorasgani, A. Farahat, K. Ristovski, C. Gupta, e G. Biswas, «A Framework for Unifying Model-based and Data-driven Fault Diagnosis», in *PHM Society Conference*, 2018, vol. 10.
- [47] K. Javed, R. Gouriveau, e N. Zerhouni, «State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels», *Mechanical Systems and Signal Processing*, vol. 94, pagg. 214–236, 2017.
- [48] C. Li, L. Li, Y. Tang, Y. Zhu, e L. Li, «A comprehensive approach to parameters optimization of energy-aware CNC milling», *Journal of Intelligent Manufacturing*, vol. 30, n. 1, pagg. 123–138, 2019.
- [49] H. Zhang, R. Kang, e M. Pecht, «A hybrid prognostics and health management approach for condition-based maintenance», in *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, 2009, pagg. 1165–1169.
- [50] A. Saxena, J. R. Celaya, I. Roychoudhury, S. Saha, B. Saha, e K. Goebel, «Designing data-driven battery prognostic approaches for variable loading profiles: Some lessons learned», in *European Conference of Prognostics and Health Management Society*, 2012, pagg. 72–732.
- [51] H. Hanachi, W. Yu, I. Y. Kim, J. Liu, e C. K. Mechefske, «Hybrid data-driven physics-based model fusion framework for tool wear prediction», *The International Journal of Advanced Manufacturing Technology*, vol. 101, n. 9–12, pagg. 2861–2872, 2019.
- [52] X. Zhang e K. A. Hoo, «Effective fault detection and isolation using bond graph-based domain decomposition», *Computers & chemical engineering*, vol. 35, n. 1, pagg. 132–148, 2011.
- [53] K. Ghosh, Y. S. Ng, e R. Srinivasan, «Evaluation of decision fusion strategies for effective collaboration among heterogeneous fault diagnostic methods», *Computers & chemical engineering*, vol. 35, n. 2, pagg. 342–355, 2011.
- [54] V. Venkatasubramanian, R. Rengaswamy, K. Yin, e S. N. Kavuri, «A review of process fault detection and diagnosis: Part I: Quantitative model-based methods», *Computers & chemical engineering*, vol. 27, n. 3, pagg. 293–311, 2003.
- [55] L. H.-T. Chiang, *Fault detection and diagnosis for large-scale systems*. University of Illinois at Urbana-Champaign, 2001.
- [56] R. G. Sargent, «Verification and validation of simulation models: an advanced tutorial», in *2020 Winter Simulation Conference (WSC)*, 2020, pagg. 16–29.
- [57] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, e K. Yin, «A review of process fault detection and diagnosis: Part III: Process history based methods», *Computers & chemical engineering*, vol. 27, n. 3, pagg. 327–346, 2003.

- [58] A. Alzghoul, B. Backe, M. Löfstrand, A. Byström, e B. Liljedahl, «Comparing a knowledge-based and a data-driven method in querying data streams for system fault detection: A hydraulic drive system application», *Computers in industry*, vol. 65, n. 8, pagg. 1126–1135, 2014.
- [59] E. Gamma, R. Helm, e R. E. Johnson, *Design patterns: elements of reusable object-oriented software*. Reading (Mass.), Etats-Unis d’Amérique, Canada, France, 1995.
- [60] C. Alexander, *A pattern language: towns, buildings, construction*. Oxford university press, 1977.
- [61] F. J. Budinsky, M. A. Finnie, J. M. Vlissides, e P. S. Yu, «Automatic code generation from design patterns», *IBM systems Journal*, vol. 35, n. 2, pagg. 151–171, 1996.
- [62] V. G. Kac, «Modular invariance in mathematics and physics», in *Proceedings of the AMS Centennial Symposium, (1992 American Mathematical Society)*, 1988, vol. 337.
- [63] V. Kreinovich e L. Longpré, «Unreasonable effectiveness of symmetry in physics», *International Journal of Theoretical Physics*, vol. 35, n. 7, pagg. 1549–1555, 1996.
- [64] E. P. Wigner, «The role of invariance principles in natural philosophy», in *Philosophical Reflections and Syntheses*, Springer, 1995, pagg. 311–320.
- [65] J. Guo, N. Zhao, L. Sun, e S. Zhang, «Modular based flexible digital twin for factory design», *Journal of Ambient Intelligence and Humanized Computing*, pagg. 1–12, 2018.
- [66] M. Dassisti e C. Semeraro, «Exergetic Control Charts (Variability Analysis in a Real Injection-moulding Industrial Application)», *Procedia CIRP*, vol. 41, pagg. 264–269, 2016.
- [67] M. Dassisti, N. Siragusa, e C. Semeraro, «Exergetic Model as a Guideline for Implementing the Smart-factory Paradigm in Small Medium Enterprises: The Brovedani Case», *Procedia CIRP*, vol. 67, pagg. 534–539, 2018.
- [68] M. Dassisti, C. Semeraro, e M. Chimenti, «Hybrid Exergetic Analysis-LCA approach and the Industry 4.0 paradigm: Assessing Manufacturing Sustainability in an Italian SME», *Procedia Manufacturing*, vol. 33, pagg. 655–662, 2019.
- [69] T. Weillkiens, *Systems Engineering with SysML/UML: Modeling, Analysis, Design*. Elsevier, 2011.
- [70] J. Poelmans, P. Elzinga, S. Viaene, e G. Dedene, «Formal concept analysis in knowledge discovery: a survey», in *International conference on conceptual structures*, 2010, pagg. 139–153.
- [71] R. Wille, «Restructuring lattice theory: an approach based on hierarchies of concepts», in *International conference on formal concept analysis*, 2009, pagg. 314–339.
- [72] M. Rouane-Hacene, M. Huchard, A. Napoli, e P. Valtchev, «Relational concept analysis: mining concept lattices from multi-relational data», *Annals of Mathematics and Artificial Intelligence*, vol. 67, n. 1, pagg. 81–108, 2013.
- [73] M. Wajnberg, P. Valtchev, M. Lezoche, A. Blondin-Massé, e H. Panetto, «FCA Went (Multi-) Relational, But Does It Make Any Difference?», *FCA4AI 2021*, pag. 27, 2021.
- [74] G. J. Williams e S. J. Simoff, *Data mining: Theory, methodology, techniques, and applications*, vol. 3755. Springer, 2006.
- [75] F. J. Venter, G. D. Oosthuizen, e J. D. Roos, «Knowledge discovery in databases using lattices», *Expert Systems with Applications*, vol. 13, n. 4, pagg. 259–264, nov. 1997, doi: 10.1016/S0957-4174(97)00047-X.
- [76] S. Andrews, «In-close, a fast algorithm for computing formal concepts», 2009.
- [77] P. Valtchev, R. Missaoui, e R. Godin, «Formal concept analysis for knowledge discovery and data mining: The new challenges», in *International Conference on Formal Concept Analysis*, 2004, pagg. 352–371.
- [78] R. Wille, «Why can concept lattices support knowledge discovery in databases?», *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 14, n. 2–3, pagg. 81–92, apr. 2002, doi: 10.1080/09528130210164161.
- [79] B. Ganter, G. Stumme, e R. Wille, *Formal concept analysis: foundations and applications*, vol. 3626. springer, 2005.
- [80] M. Lezoche e H. Panetto, «Cyber-Physical Systems, a new formal paradigm to model redundancy and resiliency», *Enterprise Information Systems*, pagg. 1–22, 2018.
- [81] N. P. Suh, D. S. Cochran, e P. C. Lima, «Manufacturing system design», *CIRP Annals*, vol. 47, n. 2, pagg. 627–639, 1998.