mathematics

Article

Manufacturing Maps, a Novel Tool for Smart Factory Management Based on Petri Nets and Big Data Mini-Terms

Javier Llopis [1,†], Antonio Lacasa [1,†], Eduardo Garcia [1,†], Nicolás Montés [2,*,†], Lucía Hilario [2,†], Judith Vizcaíno [3,†], Cristina Vilar [3,†], Judit Vilar [3,†] and Laura Sánchez [3,†] and Juan Carlos Latorre [2,†]

[1] Ford Spain, Polígono Industrial Ford S/N, Almussafes, 46440 Valencia, Spain; jllopis4@ford.com (J.L.); alacasa1@ford.com (A.L.); egarci75@ford.com (E.G.)
[2] Department of Mathematics, Physics and Technological Sciences, University CEU Cardenal Herrera, C/San Bartolome 55, Alfara del Patriarca, 46115 Valencia, Spain; luciah@uchceu.es (L.H.); juan.latorreiranzo@uchceu.es (J.C.L.)
[3] Faculty of Mathematics, University of Valencia, Blasco Ibáñez Avenue, 13, 46010 Valencia, Spain; juvizhi@alumni.uv.es (J.V.); crisvime@alumni.uv.es (C.V.); monge2@alumni.uv.es (J.V.); juvicres@alumni.uv.es (L.S.)
* Correspondence: nicolas.montes@uchceu.es
† These authors contributed equally to this work.

Abstract: This article defines a new concept for real-time factory management—manufacturing maps. Manufacturing maps are generated from two fundamental elements, mini-terms and Petri nets. Mini-terms are sub-times of a technical cycle, the time it takes for any component to perform its task. A mini-term, by definition, is a sub-cycle time and it would only make sense to use the term in connection with production improvement. Previous studies have shown that when the sub-cycle time worsens, this indicates that something unusual is happening, enabling anticipation of line failures. As a result, a mini-term has dual functionality, since, on the one hand, it is a production parameter and, on the other, it is a sensor used for predictive maintenance. This, combined with how easy and cheap it is to extract relevant data from manufacturing lines, has resulted in the mini-term becoming a new paradigm for predictive maintenance, and, indirectly, for production analysis. Applying this parameter using big data for machines and components can enable the complete modeling of a factory using Petri nets. This article presents manufacturing maps as a hierarchical construction of Petri nets in which the lowest level network is a temporary Petri net based on mini-terms, and in which the highest level is a global view of the entire plant. The user of a manufacturing map can select intermediate levels, such as a specific production line, and perform analysis or simulation using real-time data from the mini-term database. As an example, this paper examines the modeling of the 8XY line, a multi-model welding line at the Ford factory in Almussafes (Valencia), where the lower layers are modeled until the mini-term layer is reached. The results, and a discussion of the possible applications of manufacturing maps in industry, are provided at the end of this article.

Keywords: manufacturing; Petri nets; big data; Industry 4.0; smart manufacturing

MSC: 68R10; 90B15; 90B15

check for updates

Citation: Llopis, J.; Lacasa, A.; Garcia, E.; Montés, N.; Hilario, L.; Vizcaino, J.; Vilar, C.; Vilar, J.; Sánchez, L.; Latorre, J.C. Manufacturing Maps, a Novel Tool for Smart Factory Management Based on Petri Nets and Big Data Mini-Terms. *Mathematics* **2022**, *10*, 2398. https://doi.org/10.3390/math10142398

Academic Editors: Stefano Carrino, Vicente Rodríguez Montequín, Hatem Ghorbel and Ivana Budinská

Received: 8 June 2022
Accepted: 6 July 2022
Published: 8 July 2022

1. Introduction

Production, quality and maintenance are the three keys to a factory's competitiveness. In [1], an analytical study was conducted for a food company to determine the link between these three factors. When the link between maintenance and production is positive, this implies that, the more maintenance is carried out, the better the conditions and the better machines will work, enabling continuous production. When the link between quality and production is negative, this implies that quality control activities hinder production; the more hours are invested in quality control, the lower the level of production that is obtained. Although, at first glance, this seems straightforward, the model was actually difficulty to apply by the factory managers. There were many thousands of robots, clamps, cylinders, conveyor belts involved in the factory studied, plus their components, electric motors, gears, chains, which all worked in unison, requiring several dozen maintenance technicians and managers to ensure that production operated as continuously as possible. The maintenance operators sought to anticipate failure but, since failure could happen in any component, this could not be achieved perfectly, which generated high levels of stress and tension as the daily production rate had to be be maintained, as well as quality standards. For this reason, the operators and managers nick-named the factory *"The Jungle"*.

Information is essential to assist line managers and operators in their daily tasks and decision-making. Currently, factories usually have an automatic data collection system known as the FIS (factory information system), where data on the time that it takes for lines to perform their tasks are stored. Once a week, production and engineering managers download the data and process them for analysis in a meeting where engineers and managers, together with the heads of the maintenance teams, discuss the results obtained from the previous week. The objective of these meetings is to detect the greatest number of anomalies that have affected the production and quality of the product, for example, bottlenecks (CdB), causes of errors, quality failures, etc. The objective of the meeting is to propose measures that will avoid failures in the future while taking into account the cost/benefit of the measures. A key problem is that the information system does not include effective models of production lines that can facilitate decision-making. In addition, the decisions taken will not remedy losses that have already occurred, but will simply try to address the root cause.

From the point of view of maintenance, and especially predictive maintenance, data gaps can be significant. It is known that 99% of equipment failures are preceded by certain signs, conditions or indications that the failure is about to occur [2]. The best solution would be to have sensors that measure these anomalies in real time [3,4] and to be able to predict failures. However, as explained in [5], this may be very expensive since many sensors and devices would be needed. The sensors mostly used to perform CBM involve measurement of vibration, noise, oil or lubricants, electrical performance, temperature, pressure, flow, and electric consumption (see [4,6]).

Currently, there are a number of emerging approaches involving smart factory management and performance, including Industry 4.0, the Internet of Things and the Industrial Internet of Things, big data analytics, cyber-physical smart manufacturing systems, and the Data-driven Internet of Things (see [7–10]). However, with respect to predictive maintenance, all these systems have the problem of acquiring the data, that is, installing sensors and measuring the parameters of the production process. The first question asked by factory managers is how much a system like this will cost. The need to install sensors on each machine or component, with the additional cost of maintenance, multiplied by the number of components or machines to be sensorized, and considering that production, quality and maintenance must jointly be optimized, make it very difficult for industry to incorporate these approaches.

2. Previous Research: The Mini-Term—A New Paradigm for Predictive Maintenance

In [11], an improvement to existing mathematical models in the literature on production lines was proposed, and its application to improvement in the manufacturing process discussed. In the literature, data used in the analysis of manufacturing processes is classified into two types, long-term data (long-terms), and short-term data (short-terms), (see [12–16]). Long-term data are mainly used for process planning, while short-term data are mainly used for process control. Following the definition by [17], short-term data refer to a time that is not long enough for the failure period of the machine and where the cycle time of the machine is considered short-term time. In [11], short-term is redefined by two new terms, mini-term and micro-term. Mini-term can be defined as the time it takes for a part of the machine to do its job. For preventive maintenance policy or to address breakdowns, the division of a machine into mini-terms is determined by the component that could be replaced in an easier and faster way than another subdivided part of the machine. A mini-term can also be defined as a sub-division that enables understanding and investigation of machine behavior. Similarly, a micro-term can be defined as the sub-division of a mini-term (see Figure 1).

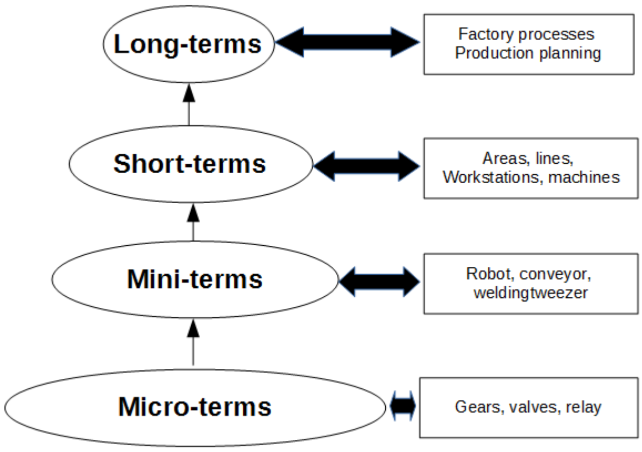

Figure 1. From the micro-term to the long-term.

The mathematical model proposed in [11] was reformulated in [18] using tensor algebra, which reduced the computational cost of the model, particularly when the number of mini-terms and micro-terms is high.

2.1. Dual Function of Mini-Terms in Maintenance and Production

The authors of [11] consider the use of mini-terms for predictive maintenance. A mini-term, by definition, is a sub-cycle time and it would only make sense to use it for production improvement. However, in [11], it was demonstrated that when a change point occurs in a conventional sensor, vibration, sound, etc., it also occurs in the mini-term, that is, a physical deterioration will also cause deterioration in the cycle time. This means that the mini-term has a dual functionality, since, on the one hand, it is a production parameter, as it measures how long it takes for the machine or component to do the task, and, on the other hand, it is the sensor used for predictive maintenance.

Figure 2 shows two examples of change points in mini-terms of a welding clamp measured at the Ford factory in Almussafes. The first change point is due to the deterioration in the proportional valve that controls the clamp movement. The second change point is due to an internal leak in the clamp cylinder. In Figure 2, it can be seen that, once the maintenance task has been performed, the time values return to their normal state.

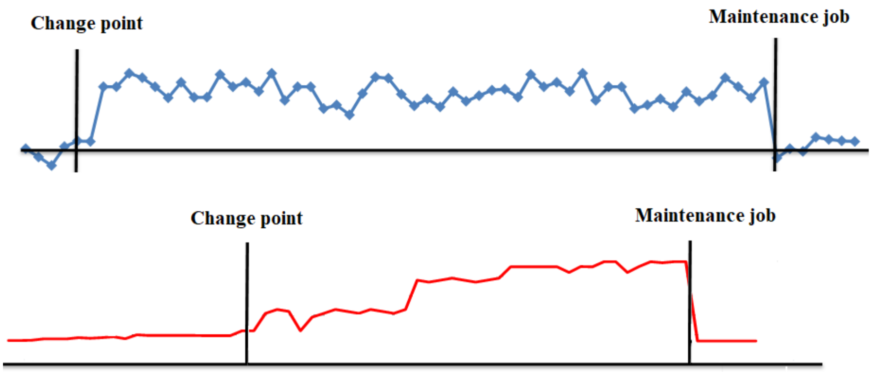

Figure 2. Change points in mini-terms. Proportional valve (**above**). Leak in cylinder (**below**).

2.2. Mini-Term 4.0. Installation Setup

As indicated in the introduction, one of the main handicaps of Industry 4.0, and other emerging approaches in the literature [7–10], is the cost of adding sensors to machines and their integration with installed systems. As also explained in the introduction, there are different prediction systems based on vibration, sound, temperature sensors, etc., but they are excessively expensive, when considering using them en masse for all the machines/components used in a factory. A large number of sensors would be required, including installation wiring, programming the measurement, etc., and for this reason, these techniques are not used extensively, except for critical machines. Portable equipment may be used instead, or a group of specialized operators may be employed to inspect machines in a sequential manner. For a real line, any component failure could cause a failure of the line, resulting in a loss of production. Thus, the success or failure of proposals related to the Industrial Internet of Things (IIOT) and Industry 4.0 is mainly influenced by the difficulty of implementation of the proposals and their economic cost.

An important advantage of using mini-terms is that no sensor, device, or additional wiring are needed, since existing components are used in the production line (controlling automatic production) for time measurement. Therefore, when measuring a mini-term, only a timer in the PLC requires to be programmed, as the existing industrial network channels the mini-terms to the database, using the mini-term as a virtual sensor (see [19]). This makes it possible to measure the mini-term, with the only cost being the programming of a timer in the PLC, enabling, for the first time within predictive maintenance, the mass monitoring of machine components and creating a new paradigm for the detection of faults in machine components [19,20]. The advantages of using mini-terms has stimulated considerable interest at the Ford Motor Company. In 2019, the company took part in a project funded by the Centre for the Development of Industrial Technology (CDIT), IDI-20190878, a public business entity dependent on the Ministry of Science and Innovation, that promotes innovation and technological development by Spanish companies. This project made it possible to implement mini-terms at the Ford Factory in Almussafes (Valencia) on a large scale and to develop the necessary algorithms for the detection of machine failures [21]. As of today, thousands of components are being monitored at the factory in Valencia through mini-terms, with a high success rate in the early detection of component failures (see [19]).

2.3. Impact of Mini-Terms on Industry

The Ford factory in Valencia was the epicentre for the implementation and testing of miniterms. The process of installing and monitoring machines and components began in 2017 when 200 mini-terms were installed and the company received a significant boost thanks to the project supported by the CDTI. Since mini-terms began to be installed and were used to predict breakdowns, different types of production indicators have begun to show significant improvements. One of the most relevant is TAV (technical availability), the percentage of planned production time without unexpected technical difficulties or maintenance needs. Figure 3 shows the development of this indicator in 2017–2020 and relates it to the number of mini-terms installed. An increase in TAV consequent on the implementation of mini-terms was confirmed.

2.4. Contribution and Paper Organization

The goal of this article is to develop a tool that allows factory employees to optimize the manufacturing process on an ongoing basis. The emergence of mini-terms, their dual utility (production and maintenance), granularity (sub-cycle time at component level) and easy implementation, represents a very valuable source of information for the governance of a factory.

This article introduces "manufacturing maps" (see Figure 4). The operation of these "maps" is very similar to that of Google maps . Google maps offers satellite imagery, aerial photography, street maps, 360º interactive panoramic views of streets ("street view"), real-time traffic conditions, and route planning for traveling by foot, car, bike and public transport. The information provided at each level of Google maps is intended for a specific use. “Manufacturing maps” are similar to Google maps but are applied to a factory. On the upper level, there is a commodity view, a complete view of the factory at a level appropriate to senior managers, and then this goes down, layer-by-layer, through a line view, a station view and, finally, a machine level view, directly involving mini-terms (see Figure 4 and 5).

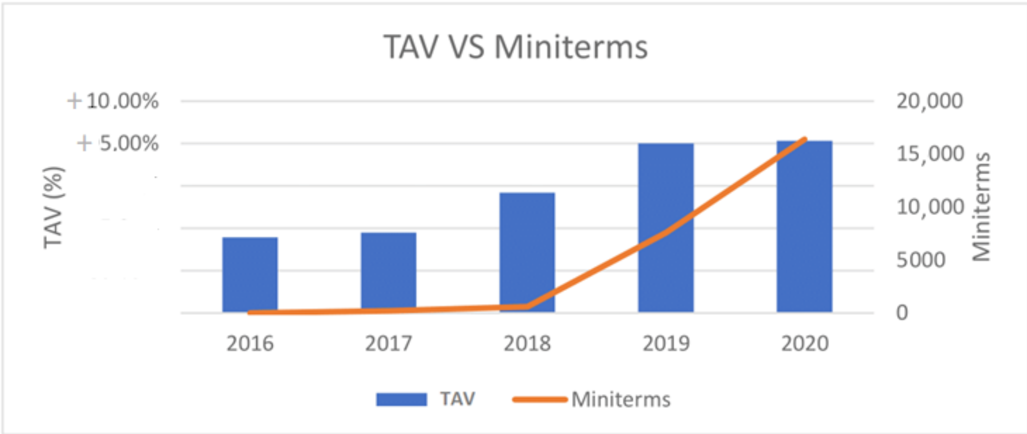

Figure 3. Improvement in production indicator (TAV) vs. mini-terms installed.

Manufacturing maps have two key components: mini-term-based big data and Petri nets. The first component provides the sub-cycle time data of the components while Petri nets provide complete factory modeling, which allows the reconstruction of information based on mini-terms at any level. To develop manufacturing maps, it is necessary to use many of the properties of Petri nets that are described in the literature, since it is necessary to develop a colored, hierarchical, time-based Petri network. The preliminary considerations for the use of Petri nets are shown in Section 3. Section 4 provides an example of Petri net design for the 8XY line from the line-view level to the lowest level where mini-terms are found. Section 5 provides an example of the application of ”manufacturing maps” that might be used by a plant manager and that enables detection of which station and model require improvements to have a direct impact on increasing production. In Section 6, we discuss the many potential applications of manufacturing maps. Section 7 provides the conclusions and considers future research directions.

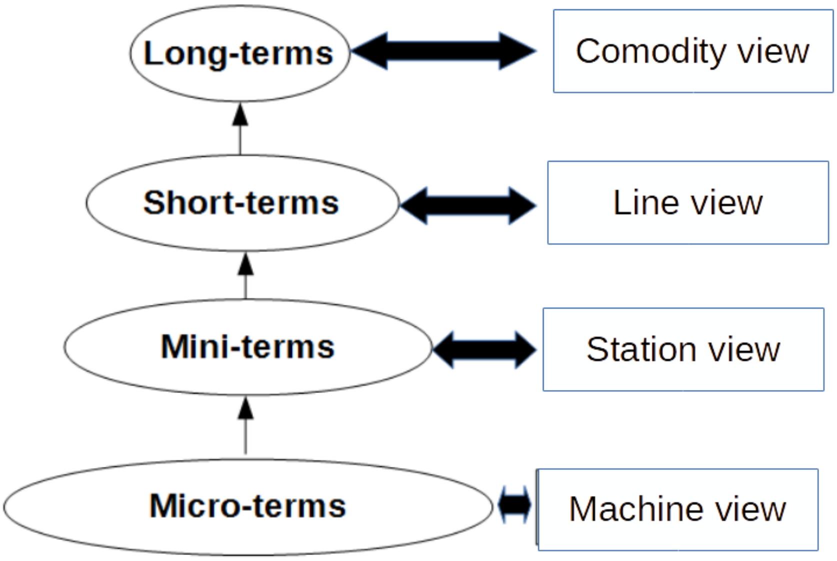

Figure 4. Data flow in manufacturing maps.

Figure 5. Manufacturing map views.

3. Previous Consideration of Petri Nets

A Petri net is a graphical or mathematical representation of a discrete event system in which the topology of a distributed, parallel or concurrent system can be described. The basic Petri net was defined in the 1960s by Carl Adam Petri. Petri nets have many applications (see [22]). For example, they can be used to model the control and data flow of automatic systems, communication protocols, synchronization systems, processor modeling, etc. At an industrial level, they also have a application in the modeling of industrial processes (see [23]). One significant application was developed in 1977 with the creation of the GRAFCET (Graphe Fonctionnel de Commande Etape Transition). The GRAFCET is a model of graphic representation of the successive behaviors of a logic system, predefined by its inputs and outputs, and is the basis of the programming of PLCs. In recent years, there has been progress in the development of different variants of Petri net networks, such as fuzzy Petri nets to model systems with uncertainty (see [24]), colored Petri nets to model processes with different products [25], a combination of both, colored fuzzy Petri nets [26] and time-based, timed colored Petri nets (see [25]).

A Petri net is represented by a bipartite directed graph. In this graph, there are two types of node:

1. Places that are the variables that define the system status. They are represented by circles.
2. Transitions that are their transformers. They are represented by a bar.

In addition, M marking is represented by a distribution in the places called tokens. A mark is represented graphically by a point inside the circle that defines the place that contains it. Places and transitions are connected by directed arcs. An arc directed from a P_i place to a T_j transition defines an input point in the transition. An output place is indicated by an arc from the transition to the place. Similarly, multiple outputs are represented by multiple arcs.

3.1. Mathematical Representation of a Petri Net

Definition 1 (Ordinary Petri net)**.** *An ordinary Petri net (RPO) [27], N, is a quadruple* $N =< P, T, Pre, Post >$*, where:*

$P = \{p_1, \cdots, p_m\}$ *is a finite and non-empty set of places;*

$T = \{t_1, \cdots, t_n\}$ *is a finite, non-empty set of transitions;*

$P \cup T = \emptyset$ *and* $P \cup T \neq \emptyset$*;*

$Pre : P \times T \rightarrow \{0, 1\}$ *is the set of input places to T;*

$post : T \times P \rightarrow \{0, 1\}$ *is the set of output places to T;*

A marked OPN is a pair $N_m =< N, M_0 >$ *in which N is an ordinary Petri net and* M_0 *is the initial marking.*

Definition 2 (Matrix representation)**.** *A PN N is defined as a matrix by means of two matrices [27]. We have* $n = |P|$ *(number of places of P) and* $m = |T|$ *(number of transitions of R). This is called:*

Pre-incidence matrix $C^- = \left[c^-_{ij}\right]_{n\times m}$ *where* $c^-_{ij} = Pre(p_i, t_j)$

Post-incidence matrix $C^+ = \left[c^+_{ij}\right]_{n\times m}$ *where* $c^+_{ij} = Post(t_i, p_j)$

Incidence matrix of N: $C = C^+ - C^-$

That is, in the incidence matrices, the rows represent the places and the columns represent the transitions. From this definition, we can represent a Petri net as follows: $RdP =< P, T, C^+, C^- >$

The matricial definition allows us to define a t_i transition by an e_i vector of dimension m (number of transitions) of components:

$$e_i(j) = \begin{cases} 1 \; si \; i = j \\ 0 \; si \; i \neq j \end{cases}$$

This matrix structure provides tools for analysing the properties and dynamic behavior of a Petri net.

3.2. Semantics of Petri Nets

The different components of a Petri net usually have the following interpretations:

- The markings represent resources in the broad sense of the word. They can be both physical resources and non-material resources.
- The resources can remain or be stored in the places.
- Transitions represent actions that consume resources to later transform them, or produce new resources.
- The weights of the arcs that go from a place to a transition represent the minimum number of resources of the class stored in said place that are needed to carry out the action represented by the transition.
- The total number of markings in a Petri net does not necessarily have to be kept because, for example, a transition can represent the assembly operation of a complex part from elementary parts, or, conversely, it can represent the disassembly of a complex part in elementary parts.

3.3. Mathematical Definition of a Hierarchical HTPN

Definition 3. *A hierarchical Petri net of* $l + 1$ *layers is a tuple of* $\{C_0, \ldots, C_{-l}\}$*, where* $C_k =< N^k_1, \ldots, N^k_{n(C_k)} >$ *for* $k = -l, -l + 1, \ldots, -1, 0$*, and* $n(C_k)$ *is the number of networks in the k layer.* N^k_j *refers to the j-th Petri net of the k layer with* $j = 1, \ldots, n(C_k)$ *that make up the set of* *places, transitions, initial marking, pre- and post-incidence matrices, and time range with a tuple structure as follows:*

$$N_j^k =< P_j^k, T_j^k, M_{0_j}^k, Pre_j^k, Post_j^k, Cis_j^k > \qquad j = 1, \ldots, n(C_k).$$

We note that, in the particular case $k = 0$ corresponding to the most superficial layer, there is only a single network and, therefore, $C_0 = \{N^0\}$.

We now consider any Petri net N of the layer k and the subnet associated with the transition i that we define as M and that is in the layer $k - 1$.

First, we have to detect from the incidence matrices which are the places of the net N that correspond to the input and output places of the net M and which, in turn, correspond to the places associated with the transition t_i. For this, we consider row i-th of the previous incidence matrix C^- of N and row i-th of the previous incidence matrix C^+ of N, respectively.

In this way, we denote by p_in of M, the set of places of N that corresponds to the positions of the row i-th of C^- not null. Similarly, we denote by p_out of M the set of places of N that corresponds to the positions of the row i-th of C^+ not null.

There are two main techniques for understanding HPN: *transition substitution* and *place substitution*.

- HPNs modeled using *transition substitution* replace transitions with subnets to implement the hierarchy. A substitution transition is a special type of transition that does not fire itself, but instead contains a subnet that defines the action or behavior that takes place. It should be noted that each substitution transition has its own subnet.
- The HPNs modeled using *place substitution* replace places with subnets to implement their hierarchy. The first step in a *place substitution* is to choose a place where the subnet will be created. Similar to the explanation provided for substitution transitions, in this case, we talk about *connecting transitions* to refer to the transitions that link a pair of places associated by hierarchy. For each connection transition in a subnet, this corresponds to one or more connection transitions in the hierarchically higher network.

3.4. Coloured Petri Nets

A colored Petri net is an extension of the mathematical concept of Petri net. The main properties of Petri nets are kept and, in this kind of Petri net, we can distinguish between one token and another. We can associate different information depending on the color to each token.

Definition 4. *A colored Petri Net (CPN) is a tuple $N =< P, T, A, C, E, \mathbb{M}, \mathbb{Y}, M_0 >$ where:*

- *P is a set of places,*
- *T is a set of transitions such that $P \cap T = \emptyset$*
- *A is a set of arcs such that $A \subset P \times T \cup T \times P$*
- *C is a color function which associates a type with each place and transition: $C : P \cup T \to \Sigma$*
- *E is a function associating an expression with each arc: $E : A \to \theta\Sigma$ with: $E(p, t), E(t, p) : C(t) \to \mu C(p)$*
- *$\mathbb{M}$ is the set of markings: $\mathbb{M} = \mu\{(p, c) : p \in P, c \in C(p)\}$*
- *$\mathbb{Y}$ is the set of steps: $\mathbb{Y} = \mu\{(t, c) : t \in T, c \in C(t)\}$*
- *$M_0 \in \mathbb{M}$ is the initial marking,*

See [28].

3.5. "Flattening" Process of Petri Nets Defined by Layers

The process of transforming an HRPN into a non-hierarchical RPN with identical behavior is referred to as a *flattening process*.

The main advantage is that we only need to test the consistency of the *flattening* process to ensure the validity of the model, because the behavior of an HRPN and its corresponding

RPN, once the *flattening* has been performed, is the same. In fact, a more optimal way to analyse and verify an HRPN is through its "flattened" representation.

The specific steps of the *flattening* process may vary for each type of Petri net. An example of the "flattening" process in a Petri hierarchical net using *transition substitution* is shown in [29].

3.6. Mathematical Definition of a TPN (Timed Petri Net)

An additional feature that can be incorporated into Petri nets is timing. This can be performed by associating time to transitions and places with the following semantics:

- When the time parameter determines the time that must elapse from when a transition is sensitized until it is fired, then proceeding to the removal and placement of markings automatically, we speak of sensitization time.
- The time parameter may also determine the time that must elapse between the removal of markings from the input places, and the placement of markings at the output places; in this case, we speak of firing time. That is, the firing of the transition has three phases (removal of input markings, firing, placement of output markings) and is not automatic, but has a duration. For this reason, this interpretation is also known as duration semantics.

We can consider the second interpretation as a particular case of the first, which can be simulated by a sequence: 'immediate firing of start transition' + 'activity in progress' + 'immediate transition firing of transition end'. Among other things, the firing time interpretation does not allow the modeling of interruptible activities, which would make it unusable for modeling real-time systems.

3.6.1. Structure of a Time Petri Net

Now, a Petri net can be defined including time in the following definition:

Definition 5. *A time Petri net (TPN), is a tuple $N =< P, T, Pre, Post, M_0, CIS >$, where:*

- $P = \{p_1, p_2, \ldots, p_n\}$ *is a set of places of cardinal n.*
- $T = \{t_1, t_2, \ldots, t_m\}$ *a set of transitions of cardinal m.*
- *Pre is the pre-incidence application, which is defined as:*

$$Pre : P \times T \to \mathbb{N}$$

- *Post is the post-incidence application, which is defined as:*

$$Post : T \times P \to \mathbb{N}$$

- $P \cap T = \varnothing$
- M_0 *is the initial marking function* $M_0 : P \to \mathbb{N}$
- *CIS is a mapping of the static intervals*
  $CIS : T \to \mathbb{Q}^+ \times (\mathbb{Q}^+ \cup \infty)$ *where* $\mathbb{Q}^+$ *is the set of positive rational numbers together with zero.*
  The last function associates to each transition a pair $(CIS(t_i) = (\alpha_i; \beta_i))$, *which defines a time interval, so it must be verified:*
  $0 \le \alpha_i < \infty$, $0 \le \beta_i < \infty$, *y* $\alpha_i \le \beta_i$ *si* $\beta_i \neq \infty$ *o* $\alpha_i < \beta_i$ *si* $\beta_i = \infty$

3.6.2. Transition Firing. Firing Rule

The Petri net works by controlling the number of tokens and how these tokens are distributed. The events in a Petri net occur by changing the distribution of tokens in places. A Petri net changes from one state to the next state when a transition fires. The flow of tokens is organized with the enabling and firing rule of a transition of a Petri net.

1. Enable rule: A transition t is said to be enabled if each input place p of t contains at least the number of tokens equal to the weight of the directed arc connecting p to t: $M(p) \geq Pre(t, p)$ for any $p \in P$
2. Firing rule: Only the enabled transition can fire. The firing of an enabled transition t removes from each input place p the number of tokens equal to the weight of the directed arc connecting p to t. It also deposits in each output place p the number of tokens equal to the weight of the directed arc connecting t to p. Firing t where M represents a new marking $M'(p) = M(p) - Pre(t, p) + Post(t, p)$ for any $p \in P$

   See [30].

3.7. Design of a Manufacturing Map for a Factory Based on Big Data Mini-Terms and Petri Nets

The design of a manufacturing map for a factory would combine the different definitions of PNs explained above and each one of these definitions would fulfil a specific function within a complex Petri net.

- Data comes from the mini-term-based big data. Mini-terms are technical sub-cycle times so the first level or network must be a time-based Petri net (TPN).
- Tokens will simulate the vehicles/models or variants to be manufactured. Depending on the line, each model may require a different treatment; for example, a model may need eight welding points of a robot but another one may need three. All this information will be included in the token and, therefore, a colored Petri net will be necessary.
- For the manufacturing map to be useful, the user must be able to choose the level to inspect/analyse or monitor. The plant manager may be interested in the higher levels while the operators or team leaders may be more interested in the lower levels which are closer to the mini-term level. Therefore, we also need to incorporate HRPNs.
- When the user selects a level above the lowest layer, the system must flatten the lower layers to display the grouped information from the lowest level.

4. Application Example. Multi-Model Welding Line 8XY

To illustrate the use of Petri nets in the construction of a manufacturing map, we will select the 8XY welding line since it is a multi-model line where 68 different models and variants are manufactured and which also has different configurations of Petri nets in the lower layers up to the layer of the mini-term. Figure 6 shows its position in the line view.

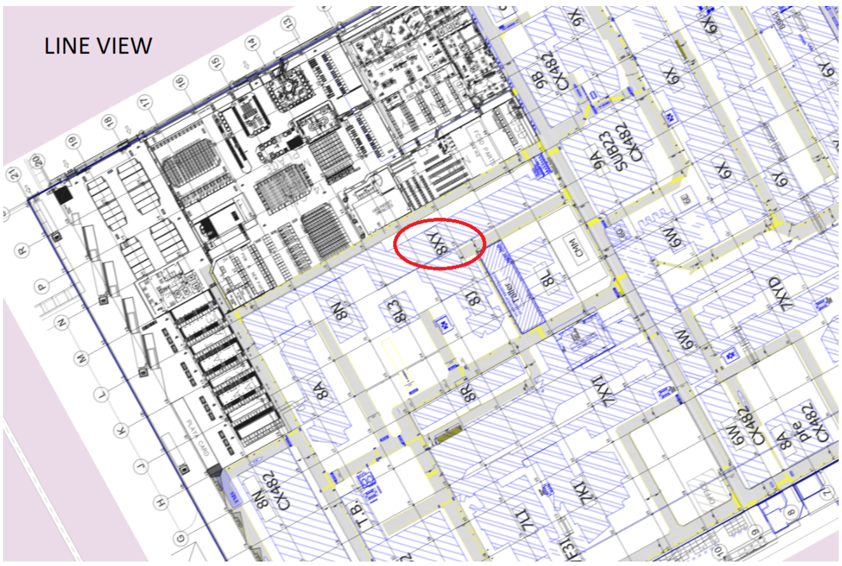

Figure 6. Red circle indicates the 8XY welding line location in the line view.

This welding line is composed of eight workstations where three of them have six welding units, four workstations with four welding units and one workstation with one welding unit (see Figure 7).

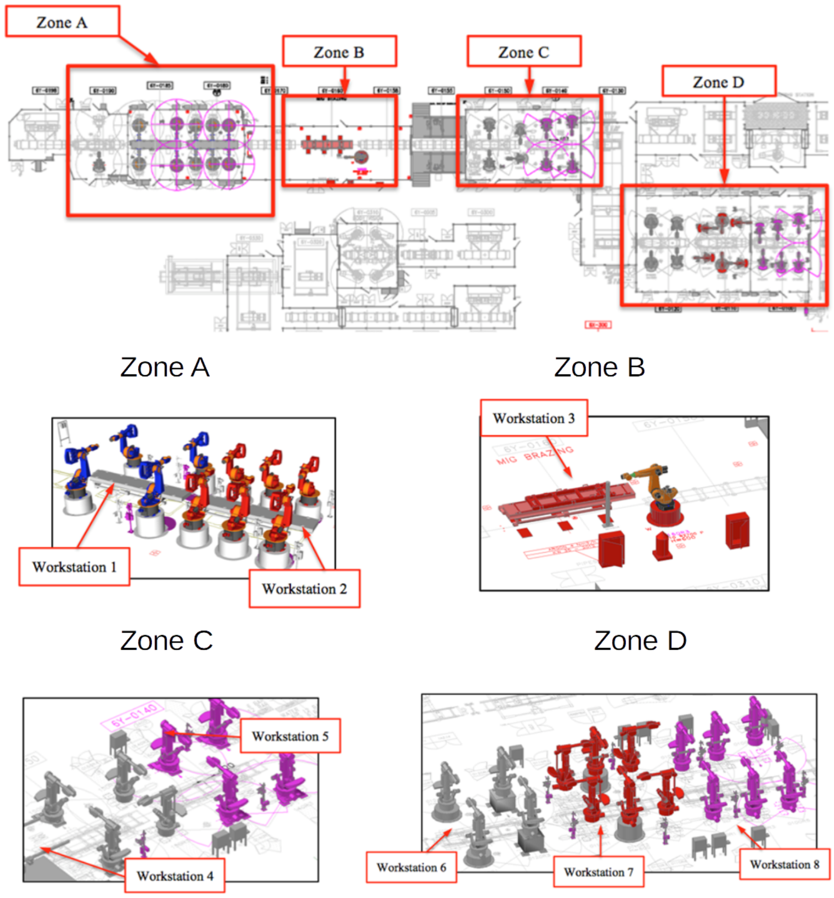

Figure 7. Layout welding line.

The welding unit consists of a robot arm and a welding clamp, see Figure 8. The behavior of the welding unit is simple. First, the robot moves the welding clamp to the point where it must be welded. Then, a pneumatic cylinder moves the clamp in two phases: first, to position it near the welding point and then to perform the action to be welded. The pressure applied by the clamp is controlled by the control system.

The robot arm and the welding clamp need a certain time to develop their task and their components also need a certain time to develop their own tasks. To analyse the deterioration effect of some mini-terms, an expert team decided that the most convenient division is in terms of three mini-terms, the robot arm, the welding clamp motion and the welding task. The reason for this division was that it is easier and faster to replace these three parts in maintenance tasks than replacing only one machine part. Therefore, in the big data of mini-terms, there are three mini-terms for each station that measure this sub-cycle time in real time. Thus, the Petri net to be built will be nourished by the real-time information of these three mini-terms: the robot arm motion, the welding clamp motion and the welding task.

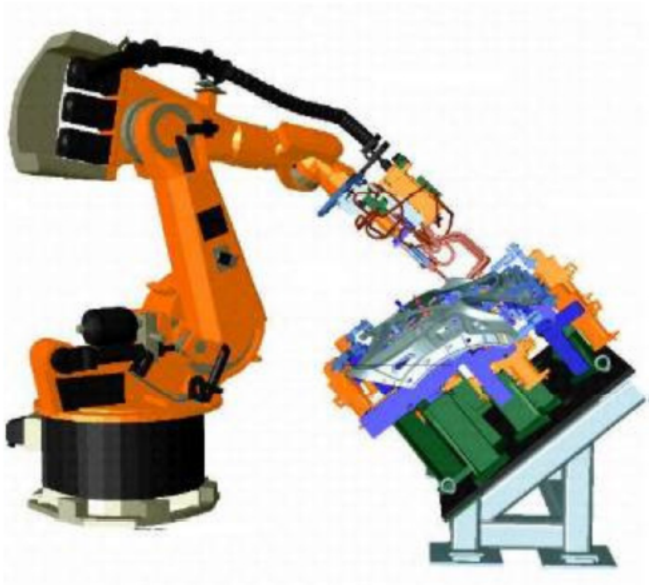

Figure 8. Welding station.

4.1. Petri Nets of Line 8XY

As an example, and without loss of generality, we model the 8XY line from the line-view plane of the manufacturing map (see Figure 6); from this level, the welding line has three layers: layer A is the outermost where we see the eight "workstations" connected in series; layer B covers each "workstation" with six, four or one welding units; finally, in layer C, the process of a welding unit divided into its mini-terms is modeled.

4.1.1. Hierarchy of Our HPN

The literature provides two ways to interpret the modeling of the process through the Petri net. The first interpretation is that each transition must be understood as an action that involves the set of activities carried out in the *workstation* from which this transition starts. In this same modeling option, we would contemplate the places as states; that is, for example, *workstation* 1 would correspond to the part introduced by the operator who waits to receive the action of the robots of the first *workstation*. Similarly, the place corresponding to the second *workstation* would correspond to the state of the part once all the robots of the first station have acted on it. In addition, in this case, we should include an auxiliary place-transition duple representing the operator who introduces the part at the beginning of the process so that everything works in order. However, another way of interpreting modeling is to consider places as actions and transitions as a place of passage between them.

In the first case, we would be talking about what is referred to in the literature as *transition substitution* (see [31,32]); that is, as transitions are understood as a set of actions, each of the transitions between *workstations* are replaced by subnets that model the operation of the set of robots corresponding to each station. However, in the second case, we would be talking about *place substitution* (see [31]), since, taking into account that what we understand as the actions are each of the places, these will be replaced by the subnets corresponding to the workstation contained in each case.

Although both possibilities are theoretically valid, from our point of view, we understand the concept of transition in a Petri net as an action. Therefore, in this application, we use an example of the concept of *transition substitution* to introduce subnets to our hierarchical network.

In this way, once we have the modeling of the main Petri net, corresponding to the specified layer A, each of the transitions is deepened into a subnet that includes all the actions carried out within the workstation corresponding to the place from which this transition starts. In this way, we have eight composite transitions replaced by subnets and one simple transition that represents the initial action of the operator introducing the part.

The subnets themselves contain subnets, which gives our hierarchical Petri net a deeper structure. In fact, in each subnet corresponding to a workstation, we find a transition including a subnet for each robot. In this way, layer −1, modeled later in a general way depending on the number of robots in each station, starts with a place of connection with the top layer; that is, the first place of layer −1, which we define by p_{in}^{-1}, is equivalent to the place of the *workstation* that represents the Petri subnet in question. Similarly, the layer −1 ends in a place, which we denote as p_{out}^{-1}, which is equivalent to the corresponding place in the network above the next station. This symbolizes the finished part in one station and moves on to the next one to receive the corresponding actions there, see Figure 9.

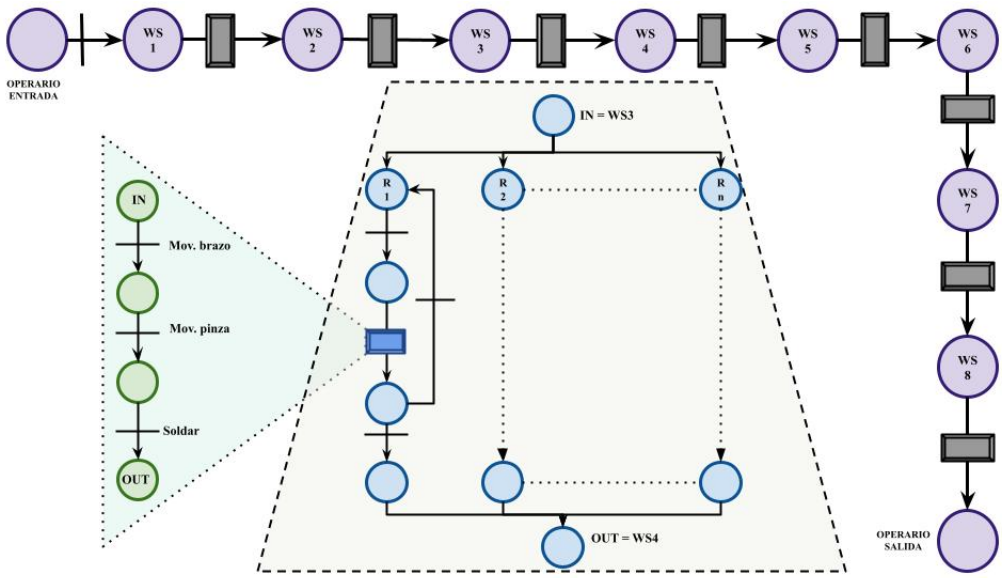

Figure 9. Hierarchical Petri net of the 8XY line.

The −2 layer results from including one subnet for each robot in the subnet of each workstation on which the tasks corresponding to this robot are performed. Keeping the thought of interpreting transitions as actions and places as states, each place represents the state of a point after an action is performed and waiting to perform the next one. As in the previous layer, layer −2 and layer −1 are related by input and output places to the lower subnet which are equivalent to places in the upper subnet. In this way, the first place of layer −2 represents the point waiting to receive the first action of the robot and is equivalent to a place in the upper subnet prior to the substituted transition that represents the point enabled for welding. The last place of layer −2 represents the welded point once the three actions of the robot have been performed on it and is equivalent in correspondence to a place of layer −1 that represents the welded point that is waiting to be sent to a "storage" place.

4.1.2. Modeling Petri Net Layer 0

We have a real welding line in which there are eight welding workstations independent of each other.

The first Petri net has as places each of the workstations and two other places, one that represents the operator including the parts and the other a place that represents the welded part, see Figure 10. We note that this is a Petri net with sequential execution, since a transition can be fired if, and only if, the previous transition is fired first.

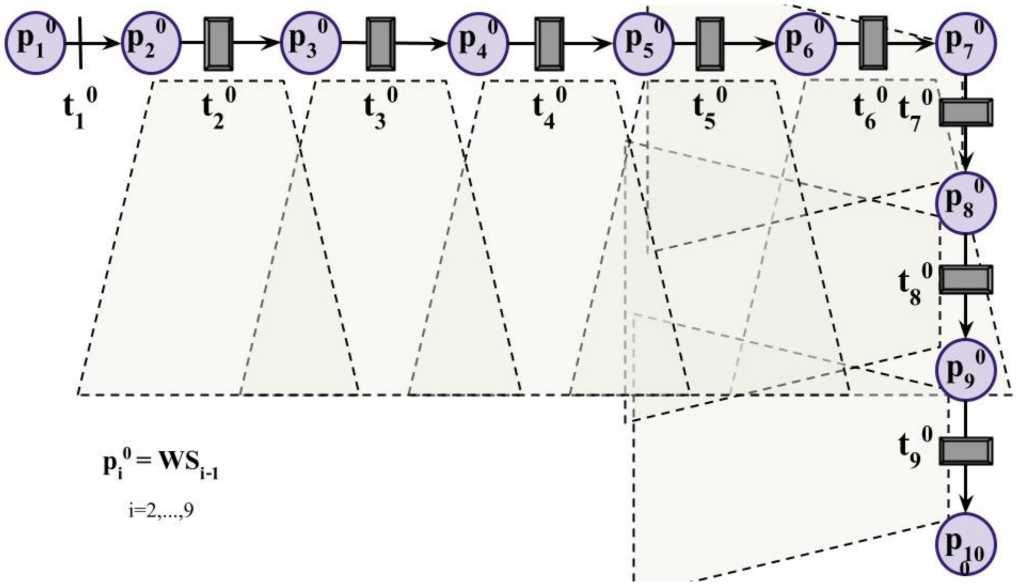

Figure 10. Modeling Petri net layer 0.

Next, we formalize the definition of the Petri net mathematically.

We consider the following sets:

- $P = \{p_1^0, p_2^0, \cdots, p_8^0, p_9^0, p_{10}^0\}$: set of places where for $i = 2, \ldots 9$ represents a workstation or *workspace*, p_1^0 represents the operator placing the part and p_{10}^0 the status of the part already welded.
- $T = \{t_1^0, t_2^0, \cdots, t_9^0\}$: set of transitions from one workstation to another. Transition t_i^0 represents the transition from p_i^0 to p_{i+1}^0.

We define the incidence applications:

- **Pre-incidence application**:

$$Pre : P \times T \to \mathbb{N}$$

- **Post-incidence application**:

$$Post : T \times P \to \mathbb{N}$$

Therefore, our Petri net is a tuple formed by $R =< P, T, Pre, Post >$.

We now define the incidence matrices of dimension 9×10 that represent the connections between the nodes of the net.

- **Pre-incidence matrix**:

$$C^-(i,j) = Pre(p_j^0, t_i^0) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = [I_9 \mid 0]$$

- **Post-incidence matrix**:

$$C^+(i,j) = Post(t_i^0, p_j^0) = \begin{bmatrix} 0&1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix} = [0 \mid I_9]$$

The **incidence matrix C** of dimension 9×10 is defined as:

$$C = C^+ - C^- = \begin{bmatrix} -1&1&0&0&0&0&0&0&0&0\\ 0&-1&1&0&0&0&0&0&0&0\\ 0&0&-1&1&0&0&0&0&0&0\\ 0&0&0&-1&1&0&0&0&0&0\\ 0&0&0&0&-1&1&0&0&0&0\\ 0&0&0&0&0&-1&1&0&0&0\\ 0&0&0&0&0&0&-1&1&0&0\\ 0&0&0&0&0&0&0&-1&1&0\\ 0&0&0&0&0&0&0&0&-1&1 \end{bmatrix}$$

Therefore, we can equivalently define the Petri net matrix as $PN =< P, T, C^+, C^- >$. On the other hand, we define each transition t_i as vectors e_i of the canonical basis; that is, a vector whose components are all 0 except the position i which is 1.

$$e_i(j) = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$$

The initial marking of the Petri net varies depending on what state it is in; that is, after a weekend when the line has been empty, the initial marking would be $M_0 = (1.0, 0, 0, 0, 0, 0)$. If the line is in operation and a line stop occurs at subsequent stations, the marking would be $M_0 = (1, 1, 1, 1, 1, 1, 1, 1, 1)$ and in any other situation it would depend on the cycle times of each model.

4.1.3. Modeling Petri Net Layer −1

In this layer, we find three *workstations* with six welding units, four *workstations* with four welding units and one *workstation* with a robot, see Figure 11. The matrix that models each zone is similar—the dimensions change depending on the number of robots in each zone since the task that each robot performs has the same representation in our PN.

The model of the layer with n robots is as follows:

- $P = \{p_1^{-1}, \ldots, p_{4n+2}^{-1}\}$
- $T = \{t_1^{-1}, \ldots, t_{4n+2}^{-1}\}$

The token enters the $p_{in}^{-1} = p_1^{-1}$ and t_1^{-1} takes a token to each robot $j, j = 1, \ldots, n$ represented by p_{4j-2}^{-1} so that they can each start their task. The transition t_{4j-2}^{-1} tries to check when welding the determined point if the robot conflicts with another one and, when it does not, then it goes to place p_{4j-1}^{-1}. At this point, it enters layer C, which we detail in the next section, and t_{4j-1}^{-1} represents the welding action of the j robot; when the token is finished, it will reach p_{4j}^{-1}, which will show the finished welded point. From this place, two transitions appear, t_{4j}^{-1} and t_{4j+1}^{-1}; the first moves the token back to p_{4j-2}^{-1} if there are more points to be welded, otherwise the second moves it to p_{4j+1}^{-1}, a waiting place until all the robots are finished. When this happens, the t_{4n+2}^{-1} transition activates and moves the token to $p_{out}^{-1} = p_{4n+2}^{-1}$ which represents that, in the "workspace" in question, the task has already been finished.

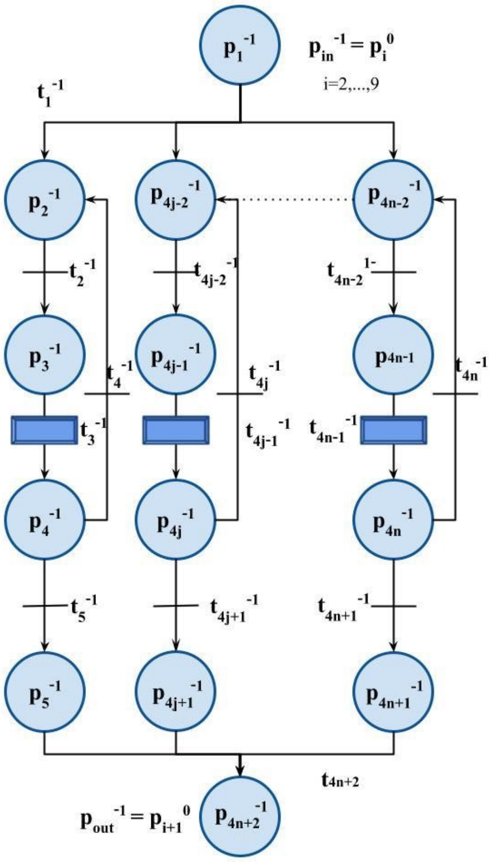

Figure 11. Modeling PN layer −1.

4.1.4. Incidence Matrices for 1 Robot

- Pre-incidence matrix for one robot

$$C^{-}(i,j) = \left[\begin{array}{c|c|c} 1 & 0 & 0 \\ ine0 & A & 0 \\ ine0 & v & 0 \end{array}\right]$$

where

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad v = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

- Post-incidence matrix for one robot

$$C^{+}(i,j) = \left[\begin{array}{c|c|c} 0 & w & 0 \\ ine0 & B & 0 \\ ine0 & 0 & 1 \end{array}\right]$$

where

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad w = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

- Incidence matrix for one robot
  Having expressed the incidence matrices by blocks of the same dimension, we can operate more easily:

$$C(i,j) = C^{+}(i,j) - C^{-}(i,j) \left[\begin{array}{c|c|c} -1 & w & 0 \\ ine0 & D & 0 \\ ine0 & -v & 1 \end{array} \right]$$

  where

$$D = B - A = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

4.1.5. Incidence Matrices for n Robots

When dealing with n robots the incidence matrices would change as follows:

- Pre-incidence matrix for n robots

$$C^{-}(i,j) = \left[\begin{array}{c|c|c|c|c|c} 1 & 0 & 0 & 0 & \cdots & 0 \\ ine0 & A & 0 & \cdots & 0 & 0 \\ ine0 & 0 & A & 0 & \cdots & 0 \\ ine0 & 0 & 0 & \ddots & 0 & 0 \\ ine0 & 0 & 0 & 0 & A & 0 \\ ine0 & v & v & \cdots & v & 0 \end{array} \right]$$

  where the matrix A and v are those of the case of a robot
- Post-incidence matrix for n robots

$$C^{+}(i,j) = \left[\begin{array}{c|c|c|c|c|c} 0 & w & w & \cdots & w & 0 \\ ine0 & B & 0 & 0 & 0 & 0 \\ ine0 & 0 & B & 0 & 0 & 0 \\ ine0 & 0 & 0 & \ddots & 0 & 0 \\ ine0 & 0 & 0 & 0 & B & 0 \\ ine0 & 0 & 0 & \cdots & 0 & 1 \end{array} \right]$$

- Incidence matrix for n robots

$$C(i,j) = C^{+}(i,j) - C^{-}(i,j) = \left[\begin{array}{c|c|c|c|c|c} -1 & w & w & \cdots & w & 0 \\ ine0 & D & 0 & 0 & 0 & 0 \\ ine0 & 0 & D & 0 & 0 & 0 \\ ine0 & 0 & 0 & \ddots & 0 & 0 \\ ine0 & 0 & 0 & 0 & D & 0 \\ ine0 & -v & -v & \cdots & -v & 1 \end{array} \right]$$

4.1.6. Modeling Petri Net Layer −2

This layer is formed by the subnet in charge of a robot welding a point so that the places are set in series, see Figure 12. The mathematical model is as follows:

- $P = \{p_1^{-2}, p_2^{-2}, p_3^{-2}, p_4^{-2}\}$
- $T = \{t_1^{-2}, t_2^{-2}, t_3^{-2}\}$

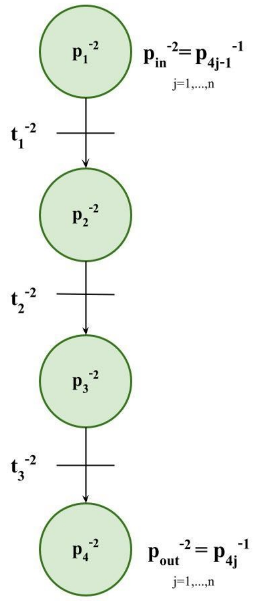

Figure 12. Modeling Petri net layer −2.

In this Petri subnet, the token will enter $p_1^{-2} = p_{in}^{-2}$ and, in the transition t_1^{-2}, the movement of the robot arm is carried out to reach p_2^{-2}, the clamp. Therefore, t_2^{-2} represents the movement of the clamp. Then the token goes to p_3^{-2} where we have the transition t_3^{-2}, which is responsible for welding the point, and ends in $p_4^{-2} = p_{out}^{-2}$ where the point is already welded.

The matrix model consists of 3×4 matrices, as follows:

- **Pre-incidence matrix**

$$C^-(i,j) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [I_3|0]$$

- **Post-incidence matrix**

$$C^+(i,j) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [0|I_3]$$

- **Incidence matrix**

$$C = C^+ - C^- = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

4.1.7. Line 8XY Tokens

In a Petri net, all tokens are black colored and the big advantage of a CPN is that each token is colored with different colors. Each type of color means a datum. The CPN tokens, which move through the PN layer 0 and through the PN layer −1, are colored tokens, and there are different tokens. The information contained in each of them refers to the number of welding points that each robot in each station must make, affecting the number of repetitions that are modeled in the PN layer −1.

5. "Manufacturing Maps" Application Example

The 8XY welding line was installed in 1980. The staff group that designed the line defined its theoretical maximum production capacity and the plant engineers defined its maximum capacity to be 51 JPH (jobs-per-hour). This value is known as the ERR (engineering running rate). This line is currently active and, since its original design, new models have been made and old models have disappeared. The introduction of a new model does not mean the complete restructuring of all the lines since other models are still being manufactured; it is, rather,a question of manufacturing the new variant in the most optimal way, minimizing the complete restructuring of all the lines and generating mismatches between the models.

The operation process at the Ford factory in Almussafes is simple. When an order for a car is placed by a customer at the dealership, it is put in the production queue to start the manufacturing process. Thus, the production queue that enters the 8XY line is organized taking into account how the orders have been placed. The 8XY line is able to manufacture 68 variants of the models manufactured by the factory in Almussafes and, obviously, through observation, we cannot detect which of the models is more difficult to manufacture and much less know where efforts should be focused for improving production capacity.

This is precisely one of the applications that manufacturing maps could have: detecting which model is the most difficult one to manufacture and where to focus the efforts of the engineering team. To do this, the "manufacturing map" has been taken from the line view of the 8XY and "flattening" has been carried out from layer -2 to 0 according to the values of the mini-terms ($\bar{x}$, S) of the last 50 samples. Manufacturing simulations have been carried out in which the 8XY line has been subjected to the manufacture of the same model continuously, obtaining a value of JPH, bias and variance, for each model. The results are shown in Figure 13.

From the results shown in Figure 13, we can see that model 44 is the model with the lowest JPH. If we repeat the simulation for that model and calculate the average waiting and blocking times, see Figure 14, we can determine where the bottleneck is using the method proposed in [13]. In this case, it would be workstation 4. With this result, the factory engineering team could focus their effort on that station and that model and improve it since any improvement made would have an immediate effect on the JPH of the line.

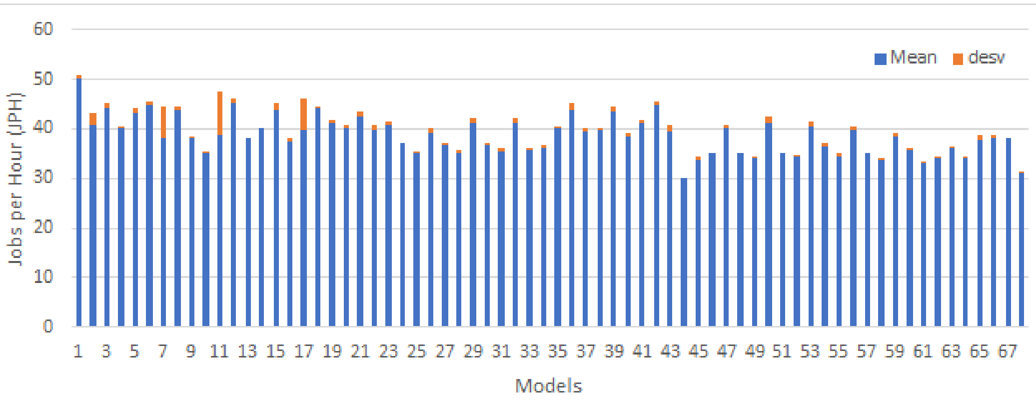

Figure 13. Jobs-per-hour for each model.

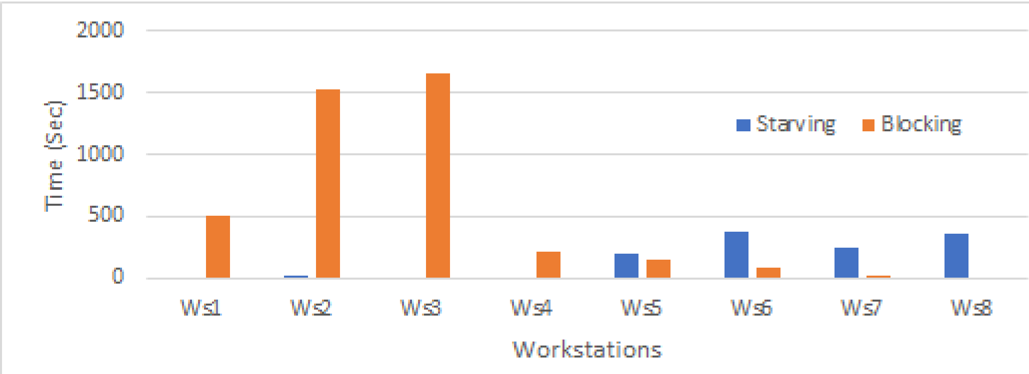

Figure 14. Starving and blocking times for each workstation.

6. Discussion

This article presents the construction of manufacturing maps using Petri nets based on big data mini-terms, which are the times that it takes for a component of the machine to perform its task. The possibility of determining the cycle times of each component in the industry in a straightforward and cheap way opens the door to a new level of management of factories since we can find out the current status of any line in any layer or at any level. If a manager wants to know the status of a particular line, he/she just needs to access the line layer and the system will calculate the flattening of the lower layers to offer the updated information for that line. This information would be accessible to any layer and, therefore, for any level of command. Senior management could choose to check higher layers and operators could focus on lower layers. The system would be able to calculate and offer the updated information as a result of performing the flattening of the lower networks that are being accessed.

Manufacturing maps based on Petri nets and mini-terms would not only represent a significant advance in the management of manufacturing lines, but would also open the door to an endless number of new applications since the simulation of Petri nets could provide valuable information to optimize the manufacturing process. An example of a potential application is shown in the previous section. However, the potential applications and improvements that manufacturing maps could offer to the industry would be enormous, for example:

- Determining real-time bottlenecks [33]. It would be possible to find out the bottleneck of each layer at each level, from the commodity level to the mini-term level, determining which component is slowing down the entire production system. This would enable scheduling of actions in these layers, calculating the benefit for each layer and determining the profitability of executing any improvement actions.
- Speed reset of robots for energy saving. Since upcoming production is known, and the times associated with each token are also known, we could estimate the waiting and blocking times the lines will have in the future. This will enable readjustment of the speed of the industrial robots, achieving considerable energy savings across the entire factory (see [34]).

7. Conclusions

This article shows how to build manufacturing maps for real-time factory management. These manufacturing maps combine the use of mini-terms in the industry and the sub-cycle time that it takes for a component to perform the task, using Petri nets modeling for the manufacturing lines. Starting from a Petri net in the lowest layer based on mini-terms, we can build Petri nets in higher layers to determine the real-time status of the line or factory. The user of a manufacturing map can select intermediate levels, such as a specific production line, and analyse or simulate with real-time data from the mini-term database. The Petri net model is built through the hierarchical network and the flattening technique, which allows building of the Petri net model for the selected level. This means that the tool has great potential for improving the management of the factory. The use of miniterms and modeling of the factory through Petri nets not only generates advantages in the management of the factory, but also opens the door to a host of new applications. Our future research will focus on developing tools such as the detection of bottlenecks by layers or the readjustment of the speed of industrial robots in real time for energy savings based on manufacturing maps.

Author Contributions: Conceptualization, J.L., A.L., E.G. and J.V. (Judith Vizcaíno); Formal analysis, A.L., E.G., L.H., J.V. (Judith Vizcaíno), C.V., J.V. (Judit Vilar) and L.S.; Funding acquisition, E.G. and N.M.; Investigation, J.L., A.L., E.G., N.M., L.H., J.V. (Judith Vizcaíno), C.V., J.V. (Judit Vilar) and L.S.; Methodology, J.L., A.L., E.G., N.M., L.H., C.V., J.V. (Judit Vilar) and L.S.; Project administration, E.G. and N.M.; Resources, E.G. and N.M.; Software, J.L., A.L. and J.C.L.; Supervision, E.G.; Validation, E.G. and J.C.L.; Visualization, J.C.L.; Writing—original draft, N.M. and L.H.; Writing—review & editing, E.G. and N.M. All authors have read and agreed to the published version of the manuscript.

Funding: Project Granted by the Valencian Innovation Agency (VIA)—Ministry of Innovation, Universities, Science and Digital Society. Action co-financed by the European Union through the Operational Program of the European Regional Development Fund (ERDF) of the Valencian Community 2014–2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RdP	Red de Petri
DOAJ	Directory of open access journals

References

1. Khan, M.R.; Darrab, I.A. Development of an analytical relation between maintenance, quality and productivity. *J. Qual. Maint. Eng.* **2010**, *16*, 341–355. [CrossRef]
2. Bosch, H.P.; Geitner, F.K. Machine failure analysis and troubleshooting. In *Technology & Engineering*; Gulf Publishing Company: Houston, TX, USA, 1983; Volume 50.
3. Peng, Y.; Dong, M.; Zuo, M.J. Current status of machine prognostics in condition-based maintenance: A review. *Int. J. Adv. Manuf. Technol.* **2010**, *50*, 297–313. [CrossRef]
4. Ahmad, R.; Kamaruddin, S. An overview of time-based and condition-based maintenance in industrial application. *Comput. Ind. Eng.* **2012**, *63*, 135–149. [CrossRef]
5. Jardine, D.A.K.S.; Banjevic, D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mech. Syst. Signals Process.* **2006**, *20*, 1483–1510. [CrossRef]
6. Kumar, S.; Goyal, D.; Dang, R.K.; Dhami, S.S.; Pabla, B.S. Condition based maintenance of bearings and gears for fault detection. A review. *Mater. Today Proc.* **2018**, *5*, 6128–6137. [CrossRef]
7. Nica, E.; Stan, C.I.; Luțan Petre, A.G.; Oașa Geambaz, R.-Ș. Internet of Things-based Real-Time Production Logistics, Sustainable Industrial Value Creation, and Artificial Intelligence-driven Big Data Analytics in Cyber-Physical Smart Manufacturing Systems. *Econ. Manag. Financ. Mark.* **2021**, *16*, 52–62.
8. Popescu, G.H.; Petreanu, S.; Alexandru, B.; Corpodean, H. Internet of Things-based Real-Time Production Logistics, Cyber-Physical Process Monitoring Systems, and Industrial Artificial Intelligence in Sustainable Smart Manufacturing. *J. Self Gov. Manag. Econ.* **2021**, *9*, 52–62.
9. Durana, P.; Perkins, N.; Valaskova, K. Artificial Intelligence Data-driven Internet of Things Systems, Real-Time Advanced Analytics, and Cyber-Physical Production Networks in Sustainable Smart Manufacturing. *Econ. Manag. Financ. Mark.* **2021**, *16*, 20–30.
10. Zvarikova, K.; Rowland, M.; Krulicky, T. Sustainable Industry 4.0 Wireless Networks, Smart Factory Performance, and Cognitive Automation in Cyber-Physical System-based Manufacturing. *J. Self Gov. Manag. Econ.* **2021**, *9*, 9–21.
11. García, E. Análisis de los Sub-Tiempos de Ciclo Técnico Para la Mejora del Rendimiento de las Líneas de Fabricación. 2016. Available online:https://repositorioinstitucional.ceu.es/handle/10637/8352 (accessed on 7 June 2022).
12. Battaia, O.; Dolgui, A. A taxonomy of line balancing problems and their solution approaches. *Int. J. Prod. Econ.* **2013**, *142*, 259–277. [CrossRef]

13. Li, L.; Chang, Q.; Ni, J. Data driven bottleneck detection of manufacturing systems. *Int. J. Prod. Res.* **2009**, *4*, 5019–5036. [CrossRef]
14. Li, L. Bottleneck detection of complex manufacturing systems using a data-driven method. *Int. J. Prod. Res.* **2009**, *47*, 6929–6940. [CrossRef]
15. Turker, A.K.; Aktepe, A.K.; Inal, A.F.; Ersoz, O.O.; Das, G.S.; Birgoren, B. Decision Support System for Dynamic Job-Shop Scheduling Using Real-Time Data with Simulation. *Mathematics* **2019**, *7*, 278. [CrossRef]
16. Thomas, T.E.; Koo, J.; Chaterji, S.; Bagchi, S. Minerva: A reinforcement learning-based technique for optimal scheduling and bottleneck detection in distributed factory operations. In Proceedings of the IEEE International Communication Systems and Networks and Workshops, COMSNET, Bengaluru, India, 3–7 January 2018.
17. Li, L.; Chang, Q.; Ni, J. Real time production improvement through bottleneck control. *Int. J. Prod. Res.* **2009**, *47*, 6145–6158. [CrossRef]
18. Garcia, E.; Montés, N. A tensor model for automated production lines based on probabilistic sub-cycle times. In *Modeling Human Behaviour: Individials and Organizations*; Chapter Book; Nova Science Pulishers: Hauppauge, NY, USA, 2017; pp. 221–234.
19. Garcia, E.; Montés, N.; Llopis, J.; Lacasa, A. Sustainable Miniterm, a novel virtual sensor for predictive maintenance for the Industry 4.0 era. *Sensors* 2022, *under review*.
20. Garcia, E.; Montés, N. Mini-term 4.0. A Real-time Maintenance Support System to Prognosticate Breakdowns in Production Lines. In Proceedings of the International Conference on Informatics in Control, Automation and Robotics, ICINCO 2019, Prague, Czech Republic, 29–31 July 2019.
21. Garcia, E.; Montés, N.; Llopis, J.; Lacasa, A. Evaluation of Change Point Detection Algorithms for Application in Big Data Mini-term 4.0.In Proceedings of the International Conference on Informatics in Control, Automation and Robotics, ICINCO 2020, Paris, France, 7–9 July 2020.
22. Murata, T. Petri Nets: Properties, Analysis and Applications. *Proc. IEEE* **1989**, *77*, 541–580. [CrossRef]
23. Zurawski, R.; Zhou, M. Petri Nets and Industrial Applications: A tutorial. *IEEE Trans. Ind. Electron.* **1994**, *41*, 567–583. [CrossRef]
24. Zhou, K.Q.; Zain, A.M. Fuzzy Petri nets and industrial applications: A review. *Artif. Intell. Rev.* **2015**, *45*, 405–446. [CrossRef]
25. Bevilacqua, M.; Ciarapica, F.E.; Giovann, M. Timed Coloured Petri Nets for Modelling and Managing Processes and Projects. *Procedia CIRP* **2018**, *67*, 58–62. [CrossRef]
26. Saren, S.; Blaga, F.; Vesselenyi, T. Implementation of fuzzy system using hierarchical Colored Petri Nets to model flexible manufacturing cell. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *400*, 42–50. [CrossRef]
27. Manuel, S. *Las Redes de Petri en la Automática y la Informática*; Editorial AC: Madrid, Spain, 1985. Available online: https://zaguan.unizar.es/record/70370/files/BOOK-2018-014.pdf (accessed on 7 June 2022).
28. Choppy, C.; Petrucci, L.; Sanogo, A. Coloured Petri Nets Refinements. PNSE'13-Petri Nets and Software Engineering.Available online: http://ceur-ws.org/Vol-989/paper11.pdf (accessed on 7 June 2022).
29. Uriel, J.; Padberg, J. Hierarchical, Reconfigurable Petri Nets. In Proceedings of the Workshops on Petri Nets and Modeling 2018, Hamburg, Germany, 11–12 October 2018. Available online: http://ceur-ws.org/Vol-2060/pemod1.pdf (accessed on 7 June 2022)
30. Wang, J. Petri Nets for Dynamic Event-Driven System Modeling. In *Handbook of Dynamic System Modeling*, 1st ed.; Chapman and Hall/CRC: London, UK, 2007.
31. Zuberek, W.M.; Bluemke, I. Hierarchies of Place/Transition Refinements in Petri Nets. In Proceedings of the 5th International Conference on Emerging Technologies and Factory Automation (ETFA'96), Kauai, HI, USA, 18–21 November 1996; pp. 355–360.
32. Huang, H.-Z.; Zu, X. Hierarchical Timed Colored Petri Nets Based Product Development Process Modeling, Xiamen, China. In *Computer Supported Cooperative Work in Design I*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3168. ISBN 978-3-540-29400-9. Available online: (accessed on)
33. Rocha, E.M.; Lopes, M.J. Bottleneck prediction and data-driven discrete-event simulation for a balanced manufacturing line. *Procedia Comput. Sci.* **2022**, *200*, 1145–1154. [CrossRef]
34. Garcia, E.; Montés, N. Real-Time idle time cancellation by means of Miniterm 4.0. *Energies* **2019**, *12*, 1207–1230. [CrossRef]