axioms

MDPI

Article

Compositional Scheduling in Industry 4.0 Cyber-Physical Systems

Fernando Tohmé [1,*] and Daniel Rossit [2]

[1] Department of Economics, Instituto de Matemática de Bahía Blanca (UNS-Conicet), Universidad Nacional del Sur, Bahía Blanca B8000, Argentina
[2] Department of Engineering, Instituto de Matemática de Bahía Blanca (UNS-Conicet), Universidad Nacional del Sur, Bahía Blanca B8000, Argentina; daniel.rossit@uns.edu.ar
* Correspondence: ftohme@criba.edu.ar

Abstract: Cyber-physical systems (CPSs) are fundamental components of Industry 4.0 production environments. Their interconnection is crucial for the successful implementation of distributed and autonomous production plans. A particularly relevant challenge is the optimal scheduling of tasks that require the collaboration of multiple CPSs. To ensure the feasibility of optimal schedules, two primary issues must be addressed: (1) The design of global systems emerging from the interconnection of CPSs; (2) The development of a scheduling formalism tailored to interconnected Industry 4.0 settings. Our approach is based on a Category Theory formalization of interconnections as *compositions*. This framework aims to guarantee that the emergent behaviors align with the intended outcomes. Building upon this foundation, we introduce a formalism that captures the assignment of operations to cyber-physical systems.

Keywords: monoidal category; hypergraph category; Industry 4.0; cyber-physical systems; smart scheduling

MSC: 18A99; 90B35; 90B36

1. Introduction

Production systems have undergone drastic changes in recent years and more radical transformations can be expected. The resulting outcome of this process has been called *Industry 4.0* [1]. It amounts to replacing traditional 20th century production systems by smart versions, in which almost all features are fully autonomous [2]. The main components of this new industrial revolution are cyber-physical systems (CPSs) that provide a direct connection between the real and the virtual world [3]. CPSs allow the real-time integration of decision-making processes with data drawn from the physical shop floor, improving the ability to respond to varying conditions [4–6]. In turn, the Internet of Things (IoT) provides the means to interconnect different CPSs yielding a single production system, able to solve decision problems, coordinate tasks, address conflicts, and plan future activities [7,8].

Achieving an efficient and coordinated integration of CPSs is not a straightforward enterprise. There are problems both at the interface layers of CPSs and at their inner workings (involving the functionalities of CPS). At the former level, problems usually arise when trying to ensure the interoperability of these systems due to compatibility failures of the communication interfaces [9]. The goal here is to ensure that information is consistently shared and communicated among all the interconnected CPSs [10]. In turn, the problems associated with the inner layers of CPS involve the emergence of unforeseen phenomena [11]. This happens when the behavior of the integrated system cannot be

check for updates

Academic Editor: Darjan Karabašević

Received: 6 March 2025
Revised: 21 April 2025
Accepted: 23 April 2025
Published: 27 April 2025

Citation: Tohmé, F.; Rossit, D. Compositional Scheduling in Industry 4.0 Cyber-Physical Systems. *Axioms* **2025**, *14*, 332. https://doi.org/10.3390/axioms14050332

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

described in terms of the behaviors of its component CPSs, that is, when the whole is greater than the sum of its parts. While emergence has its benefits, it can also lead to unintended consequences.

Thus, integrating independent CPSs poses two major challenges:

- Interoperability at the interface level, where incompatibilities in communication protocols and data formats can obstruct the flow of information.
- Emergence at the behavioral level, where the global system exhibits new dynamics that cannot be immediately inferred from its components.

In this paper, we address these issues with a formal treatment of the compositional integration of CPSs. Strict control procedures may hamper one of the main advantages of Industry 4.0 systems, namely their flexibility [1]. This implies that a careful design is needed to provide control and flexibility to systems of interconnected CPSs.

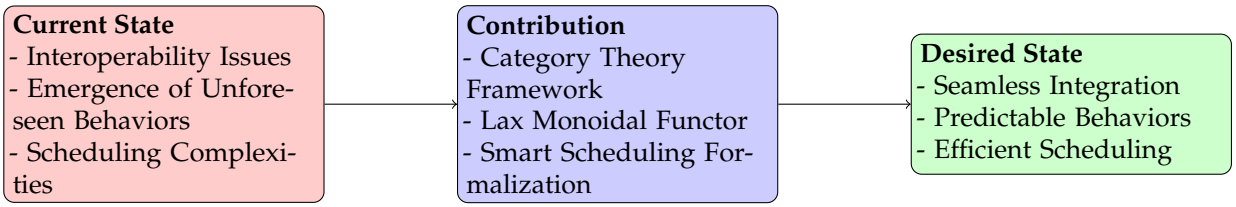
The main questions to be answered in this paper involve the assurances of compatibility of independent CPSs that can be plugged in and out of global systems. Furthermore, there is the question of the scalability of Industry 4.0 systems composed of CPSs. That is, whether plugging in a new CPS can change how a production environment works, and if so, in which way.

We consider how simple I/O systems with internal decision-making capabilities (an abstraction of the concept of cyber-physical systems) can be composed. The ensuing behavior of the larger system can be understood as the composition of the behavior of its component systems. A basic category of CPS-like units allows the definition of a symmetric monoidal structure capturing the characterization of the behavior of their composition. (a related categorical treatment of CPSs can be found in [12–14]). A *Hypergraph Category* based on this structure includes a fundamental component, namely a *lax monoidal* functor, defining an algebra of *behaviors* that parallels the composition of CPS-like systems.

One of the main applications of networks of cyber-physical systems is as platforms for Cloud Manufacturing [15–18]. This has led to the wider concept of Cyber-Physical Production Systems (CPPSs) [4,7], representing the ensemble of sub-systems connected to the environment and among them enhanced Industry 4.0 settings. One of the benefits of CPPS is the possibility of linking directly the shop floor with a higher-level Decision Support System (DSS) [8,19], providing real-time data to the DSS as well as offering the shop floor the ability to rapidly adapt to the output of the DSS.

In the case of CPPS, the intrinsic criticality of scheduling production processes becomes even more salient. As pointed out in [4], scheduling processes constitute one of the main challenges in the design of these systems. In this paper, we address this issue by showing how the lax monoidal functor of behaviors provides a way of formalizing processes of Smart Scheduling [5].

The following diagram depicts the current issues in the field, our own contribution, and what we intend to achieve.



In Section 3, we present the mathematical background of Category Theory used in the rest of the paper. Section 4 discusses the abstraction of CPS, which we call CPS-like. In Section 5, we show that networks of CPS-like components behave as a CPS-like unit itself by using the properties of a lax monoidal functor in this category. Section 6 presents

the main description of scheduling in CPPS. Section 7 uses the lax monoidal functor to characterize the solution to scheduling problems. Section 8 generalizes the framework to ensure the distributed solution of scheduling problems. Section 9 presents the conclusions.

2. Cyber-Physical Systems and Smart Manufacturing

Cyber-physical systems constitute a cornerstone in the development of Smart Manufacturing environments. They provide a means for integrating new digital capabilities into the physical processes that carry out production activities. In this section, we first show how general manufacturing and decision-making processes leverage cyber-physical technology, and second we present a detailed discussion of how scheduling decision-making processes are affected and redefined under the Industry 4.0 paradigm. After reviewing these two aspects, a final subsection summarizes the motivation of this work.

2.1. Cyber-Physical Production Systems

The Industry 4.0 paradigm focuses on the advantages of digitizing production processes to enhance the automation capabilities of the entire production environment. The key concepts that underpin the Industry 4.0 paradigm are *interconnection*, *information transparency/availability*, *technical assistance*, and *decentralized decision-making*. These four concepts enable the implementation of distributed control and the automatization of systems, enhancing the possibility of building more agile and flexible production systems [20]. The Internet of Things provides the means to interconnect the components and processes of the system, access the collected data in real time, and make them available to a decentralized architecture that benefits from them [21,22]. This data flow can be analytically processed by Big Data and Machine Learning tools, as well as with Artificial Intelligence systems, yielding meaningful descriptions of the state of the system. This, in turn, benefits from the technical assistance that Industry 4.0 systems can provide to decision-making processes. The input of the decision-making process is the information processed by these systems [23,24]. Finally, since the information is completely digitized, it can be replicated and shared by the sub-processes and sub-systems that need these data to evolve in the production process. Closing the loop, cyber-physical systems convert the output of decision-making processes into concrete actions to apply in the physical world [1,25].

This paradigm has led to the description of Industry 4.0 production systems as Cyber-physical Production Systems (CPPSs) [4]. A CPPS transforms the traditional shop floor, where the information generated by production processes is confined to its machines/assets, into a new shop floor where production machines/assets are interconnected and can be monitored and controlled remotely [26,27]. Each machine/asset has a digital representation and a Digital Twin can represent the entire shop floor. The Digital Twin can gather all the information from the shop floor and add it to the information generated by the business functions, facilitating the development of holistic decision-making processes. The information integration into Digital Twins enables the coordination and synchronization of the production process with the logistics processes and other functions [28,29]. This flow of information and the associated decision-making processes are represented in Figure 1. It depicts how the shop floor and the Digital Twin are connected and how the Digital Twin allows the autonomous configuration. Physical processes take place on the shop floor. It is where real-world data are collected (at the lowest level in Figure 1). Then, these data are transmitted in real time to the Digital Twin. This information is processed at the intermediate level of Figure 1 and fed into the decision level. The output of the decision level is a production schedule processed by the Digital Twin and enacted by the shop floor.

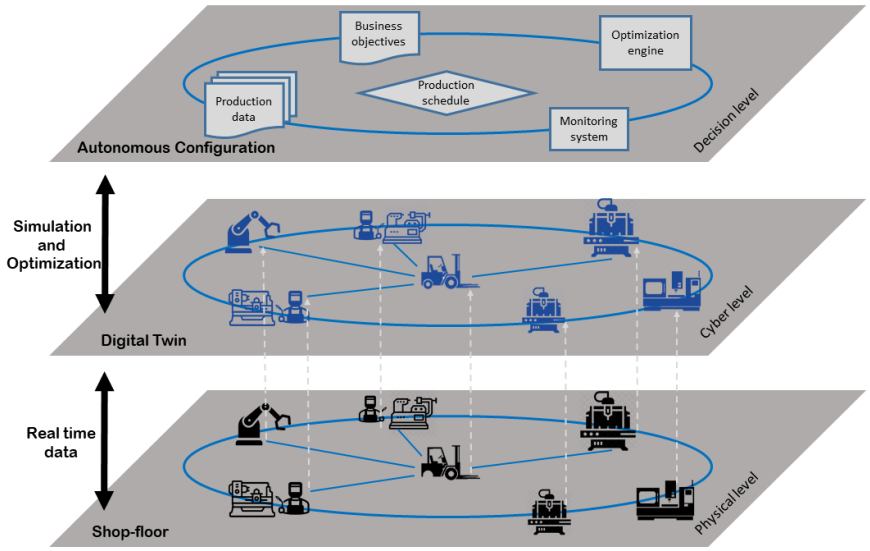


Figure 1. Smart manufacturing Digital Twin.

2.2. Scheduling Decision-Making in Cyber-Physical Production Systems

In industrial settings, the solution of scheduling decision-making problems represents the last process before starting production. Scheduling assigns different production orders to the production resources, respecting the technological restrictions and yielding a timetable of activities. Scheduling problems are solved considering the restrictions proper of the production processes (e.g., operations precedence), and also considering other features proper of industrial companies (e.g., achieving the quality specifications agreed with customers, which demand different types of setups on the machines), all this trying to optimize the performance criteria specified by the firms [30]. As expected, scheduling is carried out dynamically (i.e., the shop floor fulfills the orders as they arrive). Thus, the status of the shop floor becomes a critical input to the scheduling decision-making process. The functions of production control are closely related to this scheduling process, since any deviation from the production schedule affects the performance of the production system [31]. This shows why the topic of scheduling has been intensively studied in the context of Industry 4.0 systems in recent years [32–35].

According to this, an issue that has gained renewed interest with the emergence of Industry 4.0 is the problem of rescheduling, that is, the selection of corrective actions after disruptive (non-planned) events to diminish their impact [36]. These disruptions must be detected and analyzed in real time to take corrective actions. Naturally, since these events occur while production is being executed, the nature of the events is dynamic. The sooner these disruptions are detected and the corresponding data are collected, the better the possibility that the corrective actions may be effective. Therefore, information technologies play a central role in detecting and reacting to disruptive events. That is why CPSs leverage the capabilities of Manufacturing Execution Systems to address rescheduling problems. CPSs enable the interconnection of the entire shop floor. Every event on the floor is detected by the CPS sensors. This is translated into meaningful information, allowing the CPSs to analyze potential corrective actions in real time [1,37]. While the study of rescheduling processes has attracted the general interest of the research community [34,38,39], most of these approaches have assumed traditional scheduling processes, where the optimal solution is calculated in a centralized manner, i.e., gathering all the relevant information from all the assets/machines [40]. While these approaches take advantage of certain features

of the new information technologies, they do not use the full potential of decentralization processes [41].

2.3. Motivation

As said, rescheduling processes seek to solve those problems by gathering all the necessary data and re-optimizing decisions in a centralized way. The main argument supporting this approach is based on an integral solution to the problem, applied on the entire system and each of its sub-systems. This way of approaching the problem may not be agile enough to restrict the ability of the system to adapt to new scenarios [42]. These limitations undermine the capacities of production systems based on CPSs, which incorporate machinery according to the "plug-and-produce" logic. This article addresses this issue, providing the foundations for a possible solution of rescheduling processes in a decentralized and autonomous manner, delegating the task to each sub-system (other approaches to decentralized rescheduling are, for instance, [43–45]). To achieve this, it is vital to ensure that the decisions are made by each subsystem or element of the system (in this case CPSs), generating solutions that do not conflict with the decisions of the rest of the elements or subsystems. The decisions made by the individual components are compatible with the rest of the components and the entire system. Consequently, the *emergence* of unintended behaviors in decentralized autonomous systems must be avoided, that is, the spontaneous apparition of behaviors that were not planned. This paper proposes an approach based on Category Theory to model each of these subsystems so that decentralized decision processes do not generate the emergence of conflicting situations. Naturally, this line of study contributes to improving the decentralization and scalability of production processes, ensuring the correct interoperability of production processes based on CPS.

Consequently, his article makes the following primary contributions:

- The formulation of a Category Theory-based model for cyber-physical systems, enabling a scalable description of their interconnection.
- A theoretical framework of decision-making for rescheduling within these systems.

Our approach is purely formal, abstracting away from specific details of CPSs as in [12]. Even so, the results are intended to contribute a better understanding of these systems. The formal nature of this description may provide a schematics for possible implementations, which are not the object of the present paper.

3. Mathematical Preliminaries

Various mathematical notions are relevant to our treatment. They provide a framework for the connection of generic entities. We assume a basic knowledge of Category Theory [46].

We start with the concept of a *cospan*. Consider, in a given category $\mathcal{C}$, a central object N with input and output ports, X and Y respectively. N is called the *apex* of the cospan, while X and Y are its *feet*. The cospan can be written as $X \xrightarrow{f} N \xleftarrow{g} Y$, where f and g are morphisms between X and N and Y and N, respectively. Suppose now that another cospan shares Y with the previous one: $Y \xrightarrow{h} N' \xleftarrow{l} Z$. These cospans can be composed by connecting the two apexes to obtain a new one, with input X and output Z.

In this way, a cospan $X \xrightarrow{f} N \xleftarrow{g} Y$ can be seen as a morphism between X and Y. To be such, the apex and the feet must be objects in $\mathcal{C}$. Furthermore, we need (as the identity morphism) to define a cospan $A \xrightarrow{id} A \xleftarrow{id} A$ for each object A. This is ensured since $\mathcal{C}$ is a category.

This definition of a cospan as a morphism allows us to define a category $\mathbf{Cospan}_{\mathcal{C}}$ in which the objects are the same as in $\mathcal{C}$, while the morphisms are cospans among objects in $\mathcal{C}$. We can ensure their compositionality if pushouts are defined in the basis category $\mathcal{C}$.

Since we also want compositions in parallel, we need to be able to define a *monoidal operation* $+$ between cospans. This is enough to have both an *initial object* and *pushouts* between any pair of objects in $\mathcal{C}$.

If these conditions are fulfilled, $\mathbf{Cospan}_{\mathcal{C}}$ is a *monoidal symmetric* category, i.e., it has the following components:

- An initial object I, such that for each object A in $\mathcal{C}$ there exists a unique morphism $! : I \to A$.
- The *coproduct* $\sqcup$ in $\mathcal{C}$, provides the definition of the functor (monoidal product) $\otimes : \mathbf{Cospan}_{\mathcal{C}} \times \mathbf{Cospan}_{\mathcal{C}} \to \mathbf{Cospan}_{\mathcal{C}}$.
- For each pair of objects A and B, $A \otimes B$ is isomorphic to $B \otimes A$.

This is ensured if pushouts and an initial object exist in $\mathcal{C}$. This is the case if it is a category with *finite colimits.* An important case arises when $\mathcal{C}$ is FinSet, whose objects are finite sets. We can restrict our attention to *typed finite sets* ($\mathbf{TFS}_{\Lambda}$). We define a set Λ of types and each object in $\mathbf{TFS}_{\Lambda}$ is (X, τ_X) such that $\tau_X : X \to \Lambda$. Each X can be interpreted as a set of *ports* and τ_X indicates the type of each port. It is known that $\mathbf{TFS}_{\Lambda}$ has finite colimits [46].

We use $\mathbf{W}_{\Lambda}$ to represent $\mathbf{Cospan}_{\mathbf{TFS}_{\Lambda}}$. Its objects are called *interfaces*. The morphisms (cospans) $X \to N \leftarrow Y$ are called *wiring diagrams* and the apex N is the family of *connections*. We can thus connect three objects X^1, X^2 and X^3 to obtain a new object Y as a cospan $X^1 + X^2 + X^3 \xrightarrow{f_1} N \xleftarrow{f_2} Y$, representing the larger object defined as the network consisting of the objects X^1, X^2 and X^3.

A *Hypergraph category* [47] is a monoidal symmetric category in which the wiring diagrams constitute *networks* (that is, cables can be joined and can bifurcate). Technically, each object is equipped with a *special commutative Frobenius monoid*. In the instances to be considered here, this condition is automatically satisfied. It can be described as (Λ, H) where Λ is a class of types and $H : \mathbf{W}_{\Lambda} \to \mathbf{Set}$ is a *lax* functor such that, given an operation $\otimes$ in **Set**, there exists a morphism f such that $H(X) \otimes H(Y) \xrightarrow{f} H(X + Y)$ and, given the unit object I, there exists a morphism g such that $I \xrightarrow{g} H(\emptyset)$.

The importance of the existence of a lax functor on $\mathbf{W}_{\Lambda}$ is that in **Set** we can represent *behaviors* associated to the *structures* represented as wiring diagrams. This is particularly true in the case of $\mathbf{W}_{\Lambda}$.

An important result in this paper ensues from considering diagrams of the following form:

$$\begin{array}{ccc} \mathcal{A} & \xrightarrow{F} & \mathcal{B} \\ {\scriptstyle G}\downarrow & & \downarrow{\scriptstyle H} \\ \mathcal{C} & \xrightarrow[L]{} & \mathcal{D} \end{array}$$

where $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ are categories and F, G, H, and L are functors. If for every $a \in \mathbf{Ob}(\mathcal{A})$, there exists a unique morphism

$$H \circ F(a) \to L \circ G(a) \in \mathbf{Morph}(\mathcal{D})$$

we say that the diagram *t-commutes*.

4. A Representation of CPS

Cyber-physical systems are hard to define with precision. In particular, a definition in terms of their structure is elusive. It seems easier to provide a presentation based on their functionalities:

> Cyber-physical systems combine cyber capabilities with physical capabilities to solve problems that neither part could solve alone [48].

Examples of these range from cars to robots. They act physically on the world as determined by discrete algorithms that adjust their actuators based on sensor readings of the physical state. An abstract representation of a CPS, $X = \langle I_X, O_X, S_X, \gamma_X, \rho_X \rangle$ would thus involve the following elements:

- A set of *inputs* I_X, each of them consisting of a sensor. Each $i_X \in I_X$ has a type (for instance, a camera has a type $\mathbb{R}^{2^{\mathbb{N}}}$ representing the fact that takes two-dimensional pictures in time (recorded in discrete periods, i.e., represented by natural numbers)).
- A set of *outputs* O_X, each of them corresponding to an actuator. Again, each of them has a type, identified with its possible action (a robotic arm could make a series of movements, described by $\mathbb{R}^{3^{\mathbb{N}}}$, indicating a discrete number of positions in three-dimensional space until reaching the final position).
- A set of *internal states*, S_X, summarizing the entire information processed by the CPS.
- A function $\gamma_X : I_X \times S_X \to S_X$ and another $\rho_X : S_X \to O_X$. γ_X represents the modification of the internal state upon the reception of new inputs, while ρ_X sends the adjustments to the actuators, generating new outputs.

One of the main goals in the design of CPS is to ensure their compositional integration ([48], p. 9). In terms of Section 3, this would mean ensuring that the cyber-physical systems defined above constitute objects in a hypergraph category.

5. The Hypergraph Category of CPS

Consider a category of CPS, sharing a common set of types Λ of all its inputs and outputs, denoted $\mathcal{CPS}$. Each object X in the category corresponds to a cyber-physical system $X = \langle I_X, O_X, S_X, \gamma_X, \rho_X \rangle$.

Given two objects, $X = \langle I_X, O_X, S_X, \gamma_X, \rho_X \rangle$ and $X' = \langle I_{X'}, O_{X'}, S_{X'}, \gamma_{X'}, \rho_{X'} \rangle$, a morphism CPS

$$X \to X'$$

is such that

1. $I_X \subseteq I_{X'}$, i.e., the sensors of X' read at least the same data as those of X.
2. $O_X \subseteq O_{X'}$, meaning that $O_{X'}$ contains the same actuators as O_X.
3. $S_X \subseteq S_{X'}$, that is, the internal states of X constitute a subset of those of X'.
4. γ_X is the restriction of $\gamma_{X'}$ over $I_X \times S_X$, while ρ_X is the constraint of $\rho_{X'}$ on S_X.

Then, a morphism $X \to X'$ indicates that X is a subsystem of X'.

To complete the characterization of $\mathcal{CPS}$, we define *pushouts* and an *initial object* in this category:

- **Pushouts**: Consider three objects X, X^1 and X^2 and morphisms $X \xrightarrow{f} X^1$ and $X \xrightarrow{g} X^2$. Then, take the pushout of X^1 and X^2, denoted $X^1 + X^2$, obtained as the largest subsystem common to X^1 and X^2 such that X is one of its subsystems.
- **Initial object**: Consider the *empty system* $X^{\emptyset}$, without inputs, outputs and states. It follows trivially that $X^{\emptyset} \to X$ for every X in $\mathcal{CPS}$.

Then we have the following (the proofs of all the claims can be found in Appendix A):

Proposition 1. *$\mathcal{CPS}$ is a category with colimits.*

Since $\mathcal{CPS}$ is, according to Proposition 1, a category with colimits, we can define *cospans* in it. Consider again three objects X, X' and X'' and two morphisms $X \xrightarrow{f} X'' \xleftarrow{g} X'$. This is called a cospan from X to X'.

We can thus define the category of cospans in $\mathcal{CPS}$, denoted $\text{cospan}_{\mathcal{CPS}}$ which has a symmetric monoidal structure. Its objects are the same as those of $\mathcal{CPS}$ and a morphism $X \xrightarrow{h} X'$ is a cospan from X to X'.

Given two morphisms in $\text{cospan}_{\mathcal{CPS}}$, $X \xrightarrow{f} X'$ and $X' \xrightarrow{g} X''$, there exists a morphism $X \xrightarrow{g \circ f} X''$ that obtains as a composition of the corresponding cospans.

The monoidal structure of $\text{cospan}_{\mathcal{CPS}}$ is given by

- The unit is $X^{\emptyset}$, the initial object in $\mathcal{CPS}$.
- The monoidal product of X and X', $X + X'$, is defined as the coproduct $X \sqcup X'$.

We now present a diagrammatic language for the composition of CPS. We start by considering a symmetric monoidal category $\mathbf{W}_{\mathcal{CPS}}$. By definition, we have

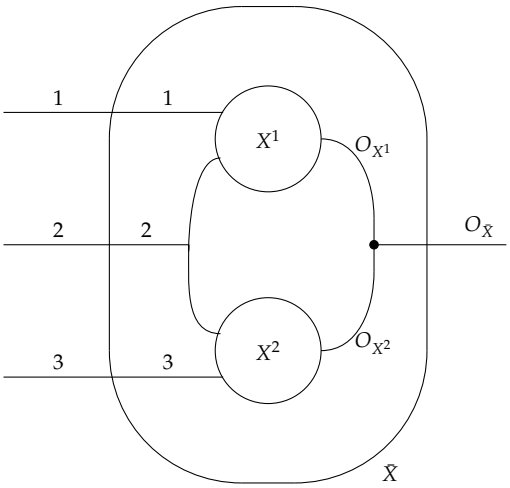
$$\mathbf{W}_{\mathcal{CPS}} = \text{cospan}_{\mathcal{CPS}}$$

Each object, i.e., a cyber-physical system X, is seen as a $\langle I_X, O_X \rangle$-labeled *interface*. Morphisms $X \to C \leftarrow X'$ yield a *wiring diagram*. The interpretation is that C is a CPS that "connects" X and X' (notice that C as an object has inputs and outputs, I_C and O_C, respectively, but to make sense, its states must be "transparent", i.e., $S_C = \emptyset$).

We write $\psi : X^1, X^2, \ldots, X_n \to \bar{X}$ to denote the wiring diagram $\phi : X^1 + X^2 + \ldots + X^n \to \bar{X}$. We can, in turn, see this as

$$X^1 + X^2 + \ldots + X^n \xrightarrow{f} C \xleftarrow{\bar{f}} \bar{X}.$$

The following figure illustrates the wiring diagram $X^1 + X^2 \to \bar{X}$ where $I_{X^1} \cap I_{X^2} = \{2\}$:



We define a *hypergraph category* $\langle \mathcal{CPS}, \text{Bhv} \rangle$ with $\text{Bhv} : \mathbf{W}_{\mathcal{CPS}} \to \bigcup_{X \in \mathcal{CPS}} S_X$, where the range represents the set of internal states in the whole family of CPS. For every object X in $\mathbf{W}_{\mathcal{CPS}}$, $\text{Bhv}(X)$ is understood as the behavior of X, encoded in its internal states.

Let us define an operation $\hat{\cup}$ such that given two internal states $s \in \text{Bhv}(X)$ and $s' \in \text{Bhv}(X')$ it yields a unique internal state in $X + X'$, combining s and s'.

Then, we have

Proposition 2. *For any pair of CPS, X and X′, Bhv*$(X)\hat{\cup}$*Bhv*$(X') = Bhv(X + X')$.

Furthermore, we also have $\emptyset = \mathrm{Bhv}(X^{\emptyset})$, where $X^{\emptyset}$ is the initial object in $\mathcal{CPS}$ and thus in $\mathbf{W}_{\mathcal{CPS}}$.

We have the following:

Proposition 3. *Bhv is a lax monoidal functor.*

6. Case Study: Scheduling Problems

Formally, a scheduling problem is the allocation of a family J of *jobs*, $J = \{1, 2, \ldots, n\}$ on a set $M = \{1, 2, \ldots, m\}$ of *machines*. Each job j consists of a class O_j of *operations*, such that O_{ij} represents an operation of job j that must be carried out on machine i.

Each operation O_{ij} has an associated pair of *starting* and *stopping* times $\langle s_{ij}, t_{ij} \rangle \in \mathbb{R}^2$ such that its processing time is $\tau_{ij} = t_{ij} - s_{ij}$. Each job j has an associated linear ordering $\prec_j$ defined on O_j, reflecting the *precedence* relation between operations. A *schedule* $\pi : J \to M$ is an assignment of jobs to machines in the following cases:

- Given O_{ij} and $O_{ij'}$, i.e., two operations on the same machine by any two different jobs j and j' on any given machine i, either $t_{ij} \leq s_{ij'}$ or $t_{ij'} \leq s_{ij}$. This means that the operations of two jobs cannot overlap on any machine.
- Given O_{ij} and $O_{i'j}$, i.e., two operations of any job j on any two different machines i and i', either $t_{ij} \leq s_{i'j}$ or $t_{i'j} \leq s_{ij}$. This means that different operations of a single job cannot be carried out simultaneously on different machines.

Denoting by F an objective function of the space Π of schedules, the *scheduling problem* is to find $\pi^* \in \Pi$ yielding the optimal value of $F(\pi)$.

Scheduling problems are highly dependent on the actual details of the production setting (*Job Shop*, *Flow Shop*, etc.). The inclusion of different parameters (*delivery dates*, *preparation times*, *waiting times*, etc.). and the adoption of different objective functions (*makespan*, *total tardiness*, *maximal tardiness*, etc.) leads to alternative statements of the general problem.

The emergence of Industry 4.0 has given rise to new capabilities that must be taken into account in the formulation of scheduling problems, particularly in rescheduling where the availability of information is a key factor. In this sense, the *Tolerance Scheduling Problem* that underpins *Smart Scheduling* becomes of special interest [5]. Tolerance Scheduling requires the analysis of the non-planned events, in such a manner that can be easily translated into scheduling parameters and variables, and to define direct rules for triggering decisions.

To address this issue, let us introduce an extra item in the description of a job j, namely its *delivery date* d_j, i.e., the time at which it should be finished. Then, given a schedule π, we can define the *tardiness* of job j as the extra time required to finish it with respect to its delivery date:

$$L_j(\pi, \mathbf{d}) \;=\; C_j(\pi) \;-\; d_j$$

where $C_j(\pi)$ is the time needed to finish job j under schedule π while $\mathbf{d}$ is the vector of delivery dates of all the jobs. Then, the maximum tardiness of the schedule is given by

$$L^*(\pi, \mathbf{d}) = \max_{j \in J} L_j(\pi, \mathbf{d})$$

In the classical version of this scheduling problem, the goal is to find the schedule $\pi^* \in \Pi$ that minimizes $L^*(\pi, \mathbf{d})$. In the case of Industry 4.0 processes carried out by a network of CPS, a perhaps more relevant goal is that of generating a range of tolerances for

the parameters of the model. Like in the specification of tolerances for manufactured goods allowing a range within which a good can still be considered appropriate, here, we allow for a certain degree of imperfection in the plan carried out.

Formally, given an optimal schedule π^* that minimizes $L^*(\pi, \mathbf{d})$, and the families of parameters $\{d_j\}_{j\in J}$ and $\{\tau_{ij}\}_{i\in M, j\in J}$, we seek a maximal interval of variations for them, incorporating an *inertia* term, δ, expressing the weight given to the stability of the system. A high δ indicates that the design favors high stability (high inertia): only a few events may trigger reschedules. That is, given an optimal schedule π^* and a class of intervals $\{[\underline{d}_j, \bar{d}_j]\}_{j\in J}$ such that for each $j \in J$, $[\underline{d}_j, \bar{d}_j] \subset \mathbb{R}$:

- Find a schedule π^{**} and $\hat{d}_j \in [\underline{d}_j, \bar{d}_j]$ for every job $j \in J$, such that
  - $L^*(\pi^{**}, \mathbf{d}) \leq L^*(\pi^*, \hat{\mathbf{d}})(1+\delta)$ and
  - $||\hat{\mathbf{d}} - \mathbf{d}|| = \max_{\{d'_j \in [\underline{d}_j, \bar{d}_j]: j\in J\}} ||\mathbf{d}' - \mathbf{d}||$.

for any schedule π. That is, the goal is to maximize the distance between the $\mathbf{d}$ parameters while ensuring that schedule π^* improves the original objective function up to an inertial factor $\delta \geq 0$. If no solution exists, a new schedule must be found under a different vector of delivery dates.

This whole process is described in Figure 2. It can be seen that the *Tolerance Scheduling problem* requires the explicit incorporation of the tolerance that planners would admit. This strategy can be used in autonomous systems.

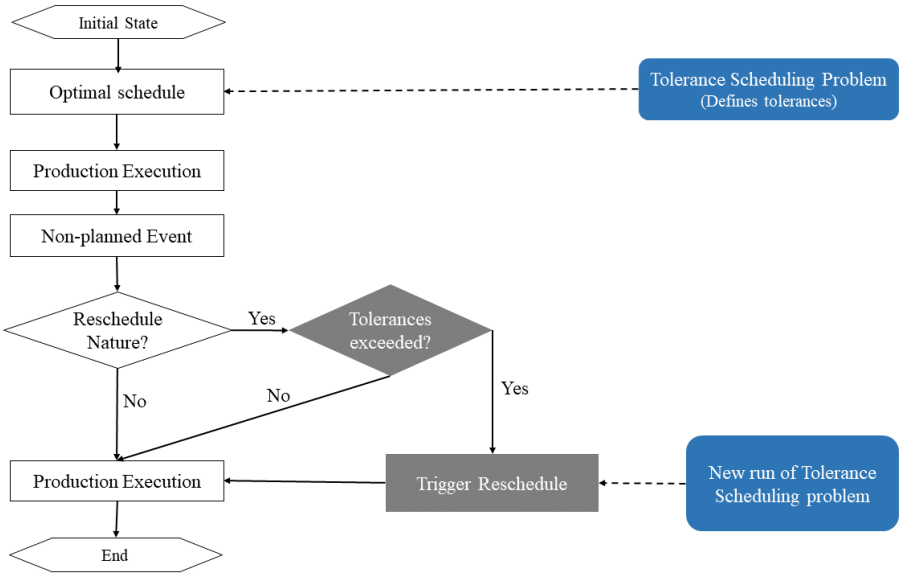


Figure 2. Smart Scheduling flow chart.

7. Categorical Scheduling

To translate Smart Scheduling to a categorical language, we start by identifying each machine in M with an object $\mathbf{W}_{\mathcal{CPS}}$. That is,

- For each $i \in M$, there exists $X^i \in \mathbf{Ob}(\mathbf{W}_{\mathcal{CPS}})$.
- A wiring diagram $\phi : X^1 + \ldots + X^m \to \bar{X}$, $\bar{X}$ represents the network of machines on which the production plan is carried out.

On the other hand, given the natural ordering of $J = \{1, \ldots, n\}$, $\preceq$, we define a trivial category $\mathcal{J}$ such that

- $\mathbf{Ob}(\mathcal{J}) = \mathbf{Ob}(J)$.
- Given $j_1, j_2 \in \mathbf{Ob}(\mathcal{J})$, there exists $f : j_1 \to j_2 \in \mathbf{Morph}(\mathcal{J})$ if and only if $j_1 \leq j_2$.

Then, $\mathcal{J} \times \mathbf{W}_{\mathcal{CPS}}$ can be defined as a category in which an object $\langle j, X^i \rangle$ can be interpreted as a job and a machine (or more in general, a network of CPS) on which an operation O_{ij} is carried out.

A morphism

$$\langle j, X^i \rangle \to \langle j^{'}, X^{i^{'}} \rangle$$

indicates that O_{ij} precedes $O_{i'j'}$.

Given a CPS X^i, its set of internal states S_{X^i} is such that any $\theta \in S_{X^i}$ is a vector $\theta = (\theta_1, \ldots, \theta_j, \ldots, \theta_n)$, where θ_j is a linear ordering of $\{O_{i,j}\}_{j \in J}$. A further condition is that θ has to satisfy the consistency conditions on the operations, avoiding overlaps and violations of precedence relations.

We can denote by J_i the set of jobs to be carried out on a CPS X^i. Then, $\bigsqcup_{i \in M} J_i$ is the direct sum of the sets of jobs corresponding to the m machines. Without loss of generality, we can assume that $J_i = J$ for each $i \in M$.

A schedule is represented by a functor

$$\pi : \mathcal{J} \to \mathbf{W}_{\mathcal{CPS}}$$

under the condition that, given $\phi : X^1 + \ldots + X^m \to \bar{X}$,

$$\mathbf{Bhv}(\bar{X}) = \hat{\bigcup}_{i \in M} \mathbf{Bhv}(X^i)$$

We denote with F^* the functor yielding the optimal value of the original objective function, pr_1 is the projection on the first component of a cartesian product, and $Int(\mathbb{R})$ is the category in which its objects are closed real intervals and the morphisms are inclusions among them.

The functoriality of all the aforementioned mappings is predicated on the assumption that, given any morphism $\langle j, X^i \rangle \to \langle j^{'}, X^{i^{'}} \rangle$, any such map (say H) is such that $H(\langle j, X^i \rangle) \to H(\langle j^{'}, X^{i^{'}} \rangle)$ since flow shop scheduling problems, which underlie Smart Scheduling problems are known to be NP-hard in the strong sense [49]. Then, on jobs, the maps preserve their natural ordering while on machines, if the complexity increases, so does the description of behaviors while tardiness does not decrease. Then, we have the following (an alternative categorical formulation of the solution of scheduling problems presents a similar condition [50]):

Theorem 1. *A schedule π is a smart schedule if the following diagram t-commutes:*

$$\begin{array}{ccc} \mathcal{J} \times \mathbf{W}_{\mathcal{CPS}} & \xrightarrow{[id_{\mathcal{J}}, \pi \circ pr_1]} & \mathcal{J} \times \mathbf{W}_{\mathcal{CPS}} \\ \downarrow{\scriptstyle [id_{\mathcal{J}}, \Phi_1]} & & \downarrow{\scriptstyle \Phi_2} \\ \mathcal{J} \times \mathbf{Bhv}(\mathbf{W}_{\mathcal{CPS}}) & \xrightarrow[(1+\delta)\mathbf{L}^* \circ \hat{\mathbf{d}}]{} & Int(\mathbb{R}) \end{array}$$

where $\Phi_1 = \mathbf{Bhv} \circ \pi \circ pr_1$ *and* $\Phi_2 = \max_j \langle C_j \circ F^* \circ \pi \rangle - \mathbf{d}$.

In particular, given a wiring diagram $\phi : X^1 + \ldots + X^m \to \bar{X}$, a *smart schedule* π assigns jobs in $\mathcal{J}$ to the CPS $X^1, \ldots, X^m$ in such a way that the corresponding $\theta \in \mathbf{Bhv}(\bar{X})$ yields a vector of tardiness $\hat{\mathbf{d}}$ that ensures that the actual optimal global tardiness belongs to an interval that is at a distance $(1 + \delta)$ of the "acceptable" value of tardiness.

8. A Generalization

The implementation of smart schedules in the category $\mathbf{W}_{\mathcal{CPS}}$ is based on distributing the information among the objects $\{X^i\}_{i\in M}$, allowing each X^i to solve its part of the problem, yielding $\mathbf{Bhv}(X^i)$. This, of course, is a consequence of the abstract nature of the X^is.

We omitted the specific details that make the objects in $\{X^i\}_{i\in M}$ different and how these distinctions induce different responses. But even taking into account those, the question remains on how the CPSs may generate π.

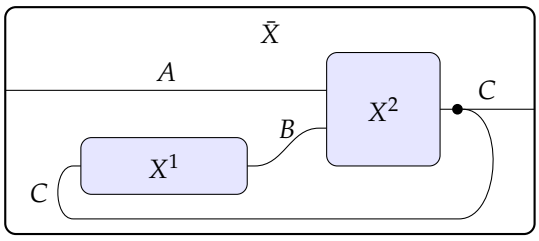
A quick initial reading may lead the reader to think that the generation of schedules must the result of a centralized process. The fact is that our formalism can be generalized to ensure the full decentralization of the decision-making procedure.

The key of this is assuming that morphisms among the X^is are now endowed with a structure given in [51] (interesting alternatives to achieve consistent decision-making procedures in distributed systems can be obtained applying *Sheaf Theory* [52] or *forcing* in the Zermelo–Fraenkel Set Theory with the Axiom of Choice [53]):

- $[X^i, X^j]$, the *internal hom* among X^i and X^j. It can be seen as an object in $\mathbf{W}_{\mathcal{CPS}}$ that summarizes the set $\{f : X^i \to X^j\}$ of morphisms between X^i and X^j.
- Given $[X^i, X^j]$, a $[X^i, X^j] - \mathbf{Coalg}$ is a category in which each object is triple $\langle s, \rho, \mu\rangle$:
  - $s \in S^{ij}$, where S^{ij} is a space of *states*, capturing the dynamics of the interaction between X^i and X^j.
  - $\rho : S^{ij} \to [X^i, X^j]$, assigning to the current state one of the input/output operations ϕ of $[X^i, X^j]$.
  - μ updates the state in response to that pattern, i.e., $\mu(\phi) = s' \in S^{ij}$.

Consider as an example a system in which two subsystems X^1 and X^2 yield a full system. X^1 has input port C and output port B, while X^2 has input ports A and B and output port C.

For any state $s \in S^{12}$ of a $[X^1, X^2]$-coalgebra (S, ρ, μ), we have $\rho(s)$ that gives a morphism $X^1 \to X^2$ in $\mathbf{W}_{\mathcal{CPS}}$ that can be depicted as



Given $(a, b, c) \in A \times B \times C$, $\mu(a, b, c)$ is the updated state in S^{12}, which in turn may yield a new connection among X^1 and X^2.

In this generalized framework, Theorem 1 still applies, but now $\Phi_2 \in [X^1 + \ldots + X^m, \bar{X}]$, arising from the distributed computation.

9. Conclusions

In this work, we presented a Category Theory-based framework for modeling cyber-physical systems (CPSs) in Industry 4.0 environments, with particular emphasis on scheduling. We addressed the problem of ensuring consistency in the composition of CPSs while providing a mathematical foundation for decentralized decision-making processes in the resulting interconnection of cyber-physical components.

We developed an abstraction of CPSs and constructed a mathematical framework that precisely captures their behavior and interactions. This offers a foundation that can

address the inherent complexity of Industry 4.0 environments. The formalization includes a symmetric monoidal structure within a basic category of CPS-like units, allowing for a precise characterization of compositional behavior. This mathematical structure enables formal reasoning about how systems behave when interconnected, providing assurances about the predictability of combined behaviors.

The categorical framework directly addresses two critical challenges identified in CPS integration:

- *Interoperability at the interface level*: the framework provides formal guarantees about the compatibility of communication interfaces when distinct CPS are connected, addressing the frequent issue of incompatible communication protocols and data formats.
- *Emergence at the behavioral level*: by formalizing the compositional integration of CPS, the framework supports emergent behaviors that arise when independent systems are combined. This is particularly relevant as emergence can lead to unintended consequences in Industry 4.0 systems.

This theoretical foundation for decentralized and autonomous decision-making is particularly suited for analyzing scheduling and rescheduling in Industry 4.0 environments. They often fail to mantain global consistency, but in our approach, production systems respond to disruptions without requiring complete system reconfiguration. More precisely, a contribution of our framework is in its capacity to prevent conflicts between decisions made by individual system components. By formalizing the relationships between component behaviors, we ensure that local optimizations do not produce globally suboptimal or conflicting outcomes. *Smart Scheduling*, characterized by the application of the *lax* monoidal functor in the category of CPSs, formalizes the design of rescheduling processes in those cases.

Future work involves a more detailed analysis of this framework, particularly in the light of real-world instances. This will contribute to the development of automated Smart Schedulers. The categorical language used here facilitates their implementation in typed languages and even in LLMs.

Author Contributions: Conceptualization, F.T. and D.R.; Methodology, F.T. and D.R.; Formal analysis, F.T. and D.R. All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

Appendix A

Proof of Proposition 1. *It is trivial to see that $\mathcal{CPS}$ is a category:*

- *For each $X \in \mathbf{Ob}(\mathcal{CPS})$, there exists the identity morphism $X \to X$ since, trivially, $I_X \subseteq I_X$, $O_X \subseteq O_X$ and $S_X \subseteq S_X$.*
- *Given two morphisms $f : X \to X'$ and $g : X' \to X''$, the composition $g \circ f : X \to X''$ obtains $I_X \subseteq I_{X'} \subseteq I_{X''}$ and thus $I_X \subseteq I_{X''}$ and analogously for O and S. Finally, γ_X and ρ_X are restrictions of $\gamma_{X'}$ and $\rho_{X'}$ which in turn are restrictions of $\gamma_{X''}$ and $\rho_{X''}$, respectively. Then, by transitivity, it follows that γ_X and ρ_X are restrictions of $\gamma_{X''}$ and $\rho_{X''}$. The associativity of $\circ$ ensues immediately.*

Finally, any category with an initial object and pushouts has all finite colimits (see Prop. 6.28 in [54]). □

Proof of Proposition 2. *If $S_X \cap S_{X'} = \emptyset$, $X + X' = X \cup X'$. Thus, each state of $X + X'$ is just the disjoint combination of states in X and X'. If, on the other hand, $S_X \cap S_{X'} \neq \emptyset$, there exists a function $H : S_{X+X'} \to S_X \times S_{X'}$ such that given $s \in S_X$ and $s' \in S_{X'}$, there exists $\bar{s} \in S_{X+X'}$ verifying that $H(\bar{s}) = \langle s, s' \rangle$.* □

Proof of Proposition 3. *If we denote by $+$ the monoidal operation in $\mathbf{W}_{\mathcal{CPS}}$ and $\otimes = \hat{\cup}$ the monoidal operation in $\prod_{X \in \mathbf{Ob}(\mathbf{W}_{\mathcal{CPS}})} S_X$, Proposition 2 indicates that there exist a trivial natural isomorphism*

$$S_X \otimes S_{X'} \to S_{X+X'}$$

Furthermore, taking the unit in $\prod_{X \in \mathbf{Ob}(\mathbf{W}_{\mathcal{CPS}})} S_X$ to be the empty set, we also have $\emptyset = S_{X^\emptyset}$, where $X^\emptyset$ is the initial object in $\mathcal{CPS}$ and thus in $\mathbf{W}_{\mathcal{CPS}}$. □

Proof of Theorem 1. *Given the jobs in $\mathcal{J}$ and the CPS $\{X^i\}_{i \in M} \subseteq \mathbf{W}_{\mathcal{CPS}}$ with a wiring diagram $\phi : X^1 + \ldots + X^m \to \bar{X}$, the lower left path in the diagram yields first the list $\{\langle j, \theta_j \rangle\}_{j \in \mathbf{Ob}(\mathcal{J})} = \mathbf{Bhv}(\pi(\mathcal{J}))$ and then the list $\{(C_j(\pi), d_j)\}_{j \in \mathbf{Ob}(\mathcal{J})}$ which allows to compute $L^*(\pi, \hat{\mathbf{d}})(1 + \delta)$ and thus the closed real interval $[0, L^*(\pi, \hat{\mathbf{d}})(1 + \delta)]$.*

On the other hand, the upper right path yields $\{\langle j, \pi(j) \rangle\}_{j \in \mathbf{Ob}(\mathcal{J})}$ and then $F^*(\pi)$ and then $L^*(\pi, \hat{\mathbf{d}})$. Then, the t-commutativity indicates that

$$[0, L^*(\pi, \hat{\mathbf{d}})] \hookrightarrow [0, L^*(\pi, \hat{\mathbf{d}})(1 + \delta)]$$

where $\hookrightarrow$ is the (unique) morphism corresponding to the inclusion among the intervals. Thus,

$$L^*(\pi, \hat{\mathbf{d}}) \leq L^*(\pi, \hat{\mathbf{d}})(1 + \delta),$$

which is a condition that should be satisfied by a smart schedule. □

References

1. Lee, J.; Bagheri, B.; Kao, H.A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manuf. Lett.* **2015**, *3*, 18–23. [CrossRef]
2. Kusiak, A. Smart manufacturing. *Int. J. Prod. Res.* **2018**, *56*, 508–517. [CrossRef]
3. Lee, E.A. Cyber physical systems: Design challenges. In Proceedings of the 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), Orlando, FL, USA, 5–7 May 2008; IEEE: New York, NY, USA, 2008; pp. 363–369.
4. Monostori, L. Cyber-physical production systems: Roots, expectations and R&D challenges. *Procedia CIRP* **2014**, *17*, 9–13.
5. Rossit, D.A.; Tohmé, F.; Frutos, M. Industry 4.0: Smart scheduling. *Int. J. Prod. Res.* **2019**, *57*, 3802–3813. [CrossRef]
6. Harkat, H.; Camarinha-Matos, L.M.; Goes, J.; Ahmed, H.F. Cyber-physical systems security: A systematic review. *Comput. Ind. Eng.* **2024**, *188*, 109891. [CrossRef]
7. Monostori, L.; Kádár, B.; Bauernhansl, T.; Kondoh, S.; Kumara, S.; Reinhart, G.; Sauer, O.; Schuh, G.; Sihn, W.; Ueda, K. Cyber-physical systems in manufacturing. *CIRP Ann.* **2016**, *65*, 621–641. [CrossRef]
8. Rossit, D.; Tohmé, F. Scheduling research contributions to Smart manufacturing. *Manuf. Lett.* **2018**, *15*, 111–114. [CrossRef]
9. Lu, Y. Industry 4.0: A survey on technologies, applications and open research issues. *J. Ind. Inf. Integr.* **2017**, *6*, 1–10. [CrossRef]
10. Romero, D.; Vernadat, F. Enterprise information systems state of the art: Past, present and future trends. *Comput. Ind.* **2016**, *79*, 3–13. [CrossRef]
11. Kopetz, H.; Bondavalli, A.; Brancati, F.; Frömel, B.; Höftberger, O.; Iacob, S. Emergence in Cyber-Physical Systems-of-Systems (CPSoSs). In *Cyber-Physical Systems of Systems*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 73–96.
12. Bakirtzis, G. Compositional cyber-physical systems theory. *arXiv* **2021**, arXiv:2109.04858.
13. Bakirtzis, G.; Vasilakopoulou, C.; Fleming, C.H. Compositional cyber-physical systems modeling. *arXiv* **2021**, arXiv:2101.10484. [CrossRef]
14. Bakirtzis, G.; Fleming, C.H.; Vasilakopoulou, C. Categorical semantics of cyber-physical systems theory. *ACM Trans. Cyber Phys. Syst.* **2021**, *5*, 32. [CrossRef]
15. Xu, X. From cloud computing to cloud manufacturing. *Robot. Comput. Integr. Manuf.* **2012**, *28*, 75–86. [CrossRef]
16. Wang, X.V.; Wang, L. From cloud manufacturing to cloud remanufacturing: A cloud-based approach for WEEE recovery. *Manuf. Lett.* **2014**, *2*, 91–95. [CrossRef]

17. Chiappa, S.; Videla, E.; Viana-Céspedes, V.; Piñeyro, P.; Rossit, D.A. Cloud manufacturing architectures: State-of-art, research challenges and platforms description. *J. Ind. Inf. Integr.* **2023**, *34*, 100472. [CrossRef]
18. Tushar, W.; Yuen, C.; Saha, T.K.; Nizami, S.; Alam, M.R.; Smith, D.B.; Poor, H.V. A survey of cyber-physical systems from a game-theoretic perspective. *IEEE Access* **2023**, *11*, 9799–9834. [CrossRef]
19. Singh, N.; Panigrahi, P.K.; Zhang, Z.; Jasimuddin, S.M. Cyber-physical systems: A bibliometric analysis of literature. *J. Intell. Manuf.* **2024**, *36*, 2335–2371. [CrossRef]
20. Xu, L.D.; Xu, E.L.; Li, L. Industry 4.0: State of the art and future trends. *Int. J. Prod. Res.* **2018**, *56*, 2941–2962. [CrossRef]
21. Da Xu, L.; He, W.; Li, S. Internet of things in industries: A survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2243.
22. Verma, R.; Koul, S.; KV, A. Evaluation and selection of a cybersecurity platform: Case of the power sector in India. *Decis. Mak. Appl. Manag. Eng.* **2023**, *7*, 209–236. [CrossRef]
23. Peres, R.S.; Jia, X.; Lee, J.; Sun, K.; Colombo, A.W.; Barata, J. Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook. *IEEE Access* **2020**, *8*, 220121–220139. [CrossRef]
24. Sanchez, M.A.; Rossit, D.; Tohmé, F. Enhancing production system resilience with digital twin-driven management. *Int. J. Comput. Integr. Manuf.* **2024**, 1–20. [CrossRef]
25. Sood, V.M.; Singh, Y.; Bhargava, B.; Narang, S.K. *Intelligent Security Solutions for Cyber-Physical Systems*; CRC Press: Boca Raton, FL, USA, 2024.
26. Pivoto, D.G.; de Almeida, L.F.; da Rosa Righi, R.; Rodrigues, J.J.; Lugli, A.B.; Alberti, A.M. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *J. Manuf. Syst.* **2021**, *58*, 176–192. [CrossRef]
27. Cao, Y.; Yang, A.; Li, H.; Zeng, Q.; Gao, J. A comprehensive knowledge map for AI improving security management of cyber-physical system enabled smart manufacturing. *Comput. Secur.* **2024**, *137*, 103650. [CrossRef]
28. Guo, D.; Zhong, R.Y.; Rong, Y.; Huang, G.G. Synchronization of shop-floor logistics and manufacturing under IIoT and digital twin-enabled graduation intelligent manufacturing system. *IEEE Trans. Cybern.* **2021**. [CrossRef] [PubMed]
29. Kaiser, J.; Mcfarlane, D.; Hawkridge, G.; André, P.; Leitão, P. A review of reference architectures for digital manufacturing: Classification, applicability and open issues. *Comput. Ind.* **2023**, *149*, 103923. [CrossRef]
30. Pinedo, M.L. *Scheduling*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 29.
31. Vieira, G.E.; Herrmann, J.W.; Lin, E. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *J. Sched.* **2003**, *6*, 39–62. [CrossRef]
32. Zhang, J.; Ding, G.; Zou, Y.; Qin, S.; Fu, J. Review of job shop scheduling research and its new perspectives under Industry 4.0. *J. Intell. Manuf.* **2019**, *30*, 1809–1830. [CrossRef]
33. Zhang, F.; Mei, Y.; Nguyen, S.; Zhang, M. Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling. *IEEE Trans. Cybern.* **2020**, *51*, 1797–1811. [CrossRef]
34. Zimmermann, E.; Mezgebe, T.T.; Haouzi, H.B.E.; Thomas, P.; Pannequin, R.; Noyel, M. Multicriteria decision-making method for scheduling problem based on smart batches and their quality prediction capability. *Comput. Ind.* **2021**, *133*, 103549. [CrossRef]
35. Villalonga, A.; Negri, E.; Biscardo, G.; Castano, F.; Haber, R.E.; Fumagalli, L.; Macchi, M. A decision-making framework for dynamic scheduling of cyber-physical production systems based on digital twins. *Annu. Rev. Control* **2021**, *51*, 357–373. [CrossRef]
36. Yilma, B.A.; Panetto, H.; Naudet, Y. Systemic formalisation of Cyber-Physical-Social System (CPSS): A systematic literature review. *Comput. Ind.* **2021**, *129*, 103458. [CrossRef]
37. Rossit, D.A.; Tohme, F. knowledge representation in Industry 4.0 scheduling problems. *Int. J. Comput. Integr. Manuf.* **2022**, *35*, 1172–1187. [CrossRef]
38. Antons, O.; Arlinghaus, J.C. Data-driven and autonomous manufacturing control in cyber-physical production systems. *Comput. Ind.* **2022**, *141*, 103711. [CrossRef]
39. Mourtzis, D.; Angelopoulos, J.; Zogopoulos, V. Integrated and adaptive AR maintenance and shop-floor rescheduling. *Comput. Ind.* **2021**, *125*, 103383. [CrossRef]
40. Uhlmann, I.R.; Frazzon, E.M. Production rescheduling review: Opportunities for industrial integration and practical applications. *J. Manuf. Syst.* **2018**, *49*, 186–193. [CrossRef]
41. Grassi, A.; Guizzi, G.; Santillo, L.C.; Vespoli, S. Assessing the performances of a novel decentralised scheduling approach in Industry 4.0 and cloud manufacturing contexts. *Int. J. Prod. Res.* **2021**, *59*, 6034–6053. [CrossRef]
42. Minguillon, F.E.; Lanza, G. Coupling of centralized and decentralized scheduling for robust production in agile production systems. *Procedia CIRP* **2019**, *79*, 385–390. [CrossRef]
43. Tan, Y.; Aufenanger, M. A real-time rescheduling heuristic using decentralized knowledge-based decisions for flexible flow shops with unrelated parallel machines. In Proceedings of the 2011 9th IEEE International Conference on Industrial Informatics, Lisbon, Portugal, 26–29 July 2011; IEEE: New York, NY, USA, 2011; pp. 431–436.
44. Sundermeier, J.; Gehlhoff, F.; Fay, A. Development of a Simulation Model to Analyze the Performance of Decentral Rescheduling Algorithms in Production Systems. *Simul. Notes Eur.* **2020**, *30*, 15–22. [CrossRef]

45. Yang, Y.; Ren, F.; Zhang, M. A Hybrid Multiagent-Based Rescheduling Mechanism for Open and Stochastic Environments Concerning the Execution Stage. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Sydney, NSW, Australia, 2–4 February 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 556–569.
46. Spivak, D.I. *Category Theory for the Sciences*; MIT Press: Cambridge, MA, USA, 2014.
47. Fong, B.; Spivak, D.I. Hypergraph categories. *J. Pure Appl. Algebra* **2019**, *223*, 4746–4777. [CrossRef]
48. Platzer, A. *Logical Foundations of Cyber-Physical Systems*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 662.
49. Garey, M.R.; Johnson, D.S. *Computers and Intractability*; Freeman: San Francisco, CA, USA 1979; Volume 174.
50. Breiner, S.; Denno, P.; Subrahmanian, E. Categories for planning and scheduling. *Not. Am. Math. Soc.* **2020**, *67*, 1666–1677. [CrossRef]
51. Tohmé, F.; Fioriti, A. Level-Agnostic Representations of Interacting Agents. *Mathematics* **2024**, *12*, 2697. [CrossRef]
52. Król, J.; Schumann, A.; Bielas, K. Brain and its universal logical model of multi-agent biological systems. *Log. Universalis* **2022**, *16*, 671–687. [CrossRef]
53. Król, J.; Schumann, A. The formal layer of {brain and mind} and emerging consciousness in physical systems. *Found. Sci.* **2023**, 1–30. [CrossRef]
54. Fong, B.; Spivak, D.I. *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*; Cambridge University Press: Cambridge, UK, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.