processes

MDPI

Article

Generic Representation Language for Modeling Transport and Material Handling Systems in Smart Manufacturing Systems

Micael Gonçalves, Paulo Martins, Guilherme Pereira and Rui Sousa *

ALGORITMI Centre, Department of Production and Systems, School of Engineering, University of Minho, 4800-058 Guimarães, Portugal; micael.goncalves@dps.uminho.pt (M.G.); pmartins@dps.uminho.pt (P.M.); gui@dps.uminho.pt (G.P.)

* Correspondence: rms@dps.uminho.pt; Tel.: +351-253510357

Abstract: This paper introduces a generic representation language to be used by organizations to represent physical and behavioral characteristics of Transport and Material Handling Systems (TMHS). This work implied a systematic observation, analysis and interpretation of several TMHS to ensure that most of the behaviors were covered. The generic representation language consists of three main types of elements: (i) objects transported, (ii) workstations and (iii) transport/handling equipment (device), and a small set of simple and easy-to-use properties to be defined by users of each organization to characterize each element of a TMHS. Each property is not related to any specific device and can be used to represent the behavior of different devices. A graphic representation for each element is proposed to make communication between users simpler and more effective, as well as to reduce the time to learn and apply the representation language. The representation of three concrete TMHS (with different behaviors, rules and restrictions) is shown, contributing to demonstrate the ability, flexibility and comprehensiveness of the developed representation language. These results point to the potential of implementing the developed generic representation language in IT (Information Technology) support systems, in particular, in Smart Manufacturing Systems, to control most of the TMHS.

Keywords: Industry 4.0; logistics; manufacturing; smart manufacturing system; transport and material handling systems

check for updates

Academic Editor: Sergey Y. Yurish

Received: 13 November 2024
Revised: 23 December 2024
Accepted: 23 December 2024
Published: 27 December 2024

Citation: Gonçalves, M.; Martins, P.; Pereira, G.; Sousa, R. Generic Representation Language for Modeling Transport and Material Handling Systems in Smart Manufacturing Systems. *Processes* **2025**, *13*, 43. https://doi.org/10.3390/pr13010043

Copyright: © 2024 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/).

1. Introduction

Industry 4.0 aims to achieve seamless connectivity among all entities in the manufacturing system, creating a Smart Factory environment where all activities are integrated and synchronized. The IT support systems responsible for managing and controlling manufacturing activities are known as Smart Manufacturing Systems [1–5]. The digital twin of manufacturing systems is the first step to implementing Smart Factories. Transport and Material Handling Systems (TMHS) are critical in implementing more efficient manufacturing practices by interacting with various entities within a manufacturing system. TMHS consists of various transport and handling equipment, including traditional equipment like tugger trains, forklifts, trucks, cranes and conveyors, as well as fully automated equipment like Automated Guided Vehicles (AGVs).

The integration of TMHS in Smart Manufacturing Systems requires further literature support [6–9]. The existing literature about representing and/or modeling TMHS is not extensive, despite its acknowledged importance. There is a need for the development of new models to represent the physical and information flow of manufacturing systems. Most

reviewed studies focus on introducing new advanced equipment [10], optimizing performance indicators [11–15] and identifying waste for improvements [8,16]. The characteristics of TMHS have been changing over time, and the activities of transport equipment cannot be controlled with existing models. Some challenges have been identified in applying existing techniques related to synchronization, comprehensiveness, integration, communication and scope. Existing models do not represent the interaction between different transport equipment or between different shop floor entities (such as execution workstations, warehouses and supermarkets), do not consider handling activities, are not prepared to integrate with other Manufacturing Planning and Control Systems, are not prepared to communicate with external systems and equipment and do not apply to different organizations and types of industry [3,7,10,17,18]. As a result, companies often develop small applications for specific scenarios. However, they are aware that this approach is not ideal because it requires high development and maintenance costs, as well as constant technical assistance [3,17]. A research gap is thus verified, justifying the relevance of the work proposed in this paper.

The development of a model capable of representing and controlling different TMHS to integrate into a Smart Manufacturing System is a complex exercise. The representation of TMHS is challenging because they can have different elements, behaviors, rules and restrictions. The main challenges in representing TMHS stem from the variety of transport and handling equipment, the several places that may be visited, the different load transfer processes for equipment, the storage areas of workstations, the types of objects handled and transported and the communication with industrial equipment [3]. To overcome this problem, it is necessary to have rules to represent TMHS, capable of fitting the requirements of different transport and handling equipment, so that, based on the information represented, processes can be developed to control transport and handling activities and integrate with the remaining systems [19].

This paper introduces a generic language for organizations to represent the physical and behavioral characteristics of TMHS in a simple and efficient manner. The language consists of three element types and a small set of properties that users can apply to represent devices and activities within discrete manufacturing systems. The language does not apply to TMHS implemented in continuous manufacturing systems since the requirements may not be the same and they have not been considered in the scope of this work. Transport equipment used in continuous manufacturing systems, such as a water pipeline, are usually in constant operation and transport identical products, while discrete manufacturing systems are characterized by a high diversity of equipment and products. A device can refer to a single piece of equipment or a group of equipment, such as a set of AGVs with the respective fleet management platform. A graphic representation for each element is proposed to create a clear and objective language and standardize the communication between different users. This work is part of an ongoing research project that aims to develop a generic model for representing and controlling TMHS in Smart Manufacturing Systems. This work follows the action research methodology [20].

The paper is structured in six sections. Section 2 describes the methods to carry out this work. Section 3 presents the element types and properties of the representation language. Section 4 shows the main results of applying the proposed language to represent different devices and TMHS. Section 5 discusses the main results and limitations. Section 6 sums up the main conclusions and future work of the project.

2. Methods

The development of the representation language involved five key phases (Figure 1). Firstly, a state-of-the-art review and data collection from industrial case studies was conducted to gain a comprehensive understanding of the topic and to define the objectives and

requirements of the representation language. The research provided hundreds of results and, to find the most relevant documents, the papers were limited to the subject areas of engineering and computer science, and reviewed based on the titles, abstracts, table of contents and conclusions.

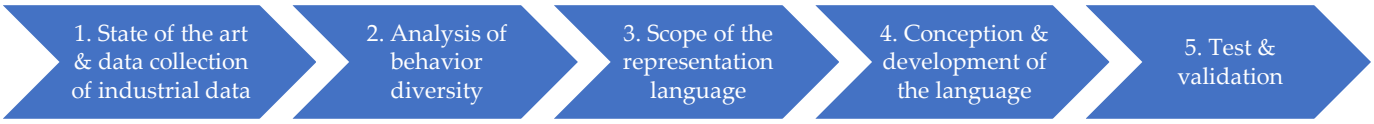


Figure 1. Overview of the research methodology.

Following this, the analysis of behaviors that characterize TMHS was carried out, involving systematic observation, description and interpretation of different examples of devices and TMHS. The third phase focused on defining the scope of the representation language to establish the problem's boundaries by studying and interpreting the behaviors of each device and TMHS. Subsequently, the representation language was developed by defining the set of element types and properties capable of representing the behaviors identified in the previous phase. Finally, selected case studies were used to assess and validate the representation language, with testing conducted on real industrial devices. This phase led to new developments or adjustments in the representation language based on the results (phase 4).

3. Generic Representation Language

The generic representation language presented in this paper defines a set of element types and properties that organizations can use to represent TMHS. A graphic representation for each element is proposed whenever applicable. Figure 2 shows three stages of the representation language, considering Reference Architectural Model Industrie 4.0 (RAMI 4.0), to "preview" the outcomes of its application in industrial organizations [21]. The three stages are as follows:

- Representation language: element types and properties identified in this paper to be used by different organizations to represent each element of the TMHS (Axis "Life Cycle & Value Stream" of RAMI 4.0—Type);
- TMHS: representation of the TMHS of a specific organization (defined by users of the organization)—to be used by an application to control the activities of each device. The representation uses the element types and properties available in the representation language and respects its syntax rules (Axis "Life Cycle & Value Stream" of RAMI 4.0—Instance);
- Application: combines the representation of the TMHS with the data of the manufacturing system at each moment to control the activities of each device, through the development of processes to schedule and control transport and handling activities and their implementation in an IT support system or a Smart Manufacturing System (Axis "Layers" of RAMI 4.0—Asset to Business).

This section is divided into three subsections that refer to different element types of TMHS—the objects transported, the places in a manufacturing system and the devices in a TMHS. Each subsection presents the definition of each element type and property, and some examples of the definition of each element are also shown to explain how the representation language can be used by organizations.

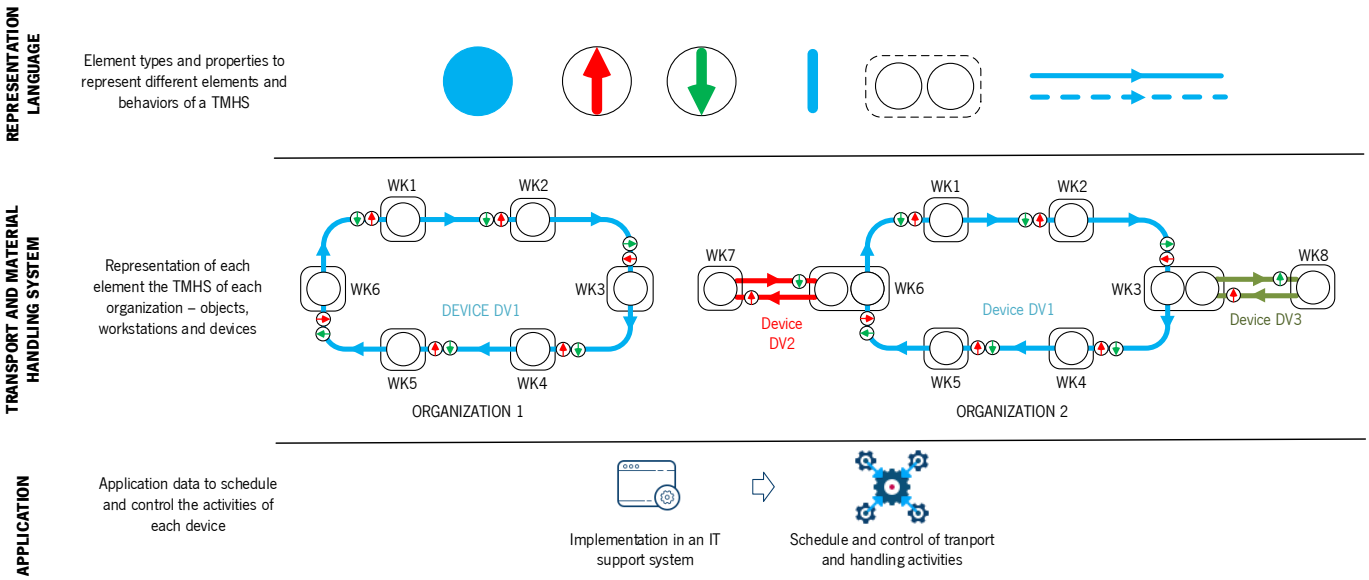


Figure 2. Stages of the representation language.

3.1. Object

An **object** is, in the representation language, a broad term for anything that a device can transport, including materials, machine components, tools and equipment. In the context of this research project, a device can transport two different objects—volumes or storage locations.

A **volume** refers to a quantity of items that can be handled and transported between two places to meet the needs of the manufacturing system. It can refer to a single item, a set of materials, a batch or a container.

A **storage location** refers to a unit that holds a set of volumes, such as pallets, dollies or a storage rack, which can be transported. When a storage location is transported, all volumes placed in the storage location are also transported, regardless of whether all volumes are required at the destination or not.

3.1.1. Volume Parameter

Manufacturing systems handle a wide variety of volumes and items, with differing weights and shapes. A **volume parameter** is an attribute created by organization users to characterize a volume. Examples of volume parameters include the type of product in a volume, the shape or the weight. Each volume parameter (VP) is characterized by the following properties [19]:

- Volume parameter ID [VP_{ID}]: unique identifier of the volume parameter;
- Parameter domain [VP_{DOMAIN}]: domain of the volume parameter. The domain of each parameter is the set of values accepted to define the parameter—which can be defined by comprehension or extension. For sets defined by extension, the organization must define all possible values ('Value 1', 'Value 2', etc.).

Example: Table 1 shows an example of the definition of two volume parameters—VP1 (product type) and VP2 (weight). VP1 is defined by extension, and its value domain consists of three values—'Raw Material', 'Semi-Finished' and 'Final Good'. VP2 is defined by comprehension, and its value domain considers any integer value between 1 and 9999.

Table 1. Example of the definition of volume parameters.

Volume Parameter ID	Parameter Domain
VP1	'Raw Material' 'Semi-Finished' 'Final Good'
VP2	[1–9999]

3.1.2. Storage Location Parameter

A **storage location parameter** is an attribute created and defined by the organization to characterize a storage location. Each storage location parameter (SLP) is defined by the following properties [19]:

- Storage location parameter ID [SLP_{ID}]: unique identifier of the SLP;
- Parameter domain [SLP_{DOMAIN}]: domain of the storage location parameter.

3.2. Workstation

The primary role of devices in a TMHS is to handle and transport objects to different workstations. A **workstation** is any place in a manufacturing system where a device can go and where a device can execute an activity, such as execution workstation, warehouses, transfer points or external entities. Although workstations have different behaviors within a manufacturing system, for transport and handling devices, they all serve a similar purpose—a place where a device can go and execute an activity [3]. Each workstation (WK) is characterized by the following properties [19]:

- Workstation ID [WK_{ID}]: unique identifier of the workstation;
- Workstation type [WK_{TYPE}]: property to define different rules for transferring objects at the workstation WK_{ID}. The value domain of this property is {'Workstation', 'Transfer point' or 'Meeting point'} (the behavior of each value is defined below);
- Physical location [$WK_{LOCATION}$]: physical location in the manufacturing system. The value of the property can be identified, for example, based on an internal system or GPS coordinates. This property provides information that can be used to send instructions to equipment (for instance, a coordinate of a workstation that should be visited by a Self-driving vehicle).

Workstation type [WK_{TYPE}]: A 'Workstation' is a classification for a workstation to where objects can be transported based on the requirements of manufacturing systems (but not transferred between devices).

A 'Transfer point' is a workstation where objects can be unloaded from one device and then loaded into another device. At a transfer point, a device can unload an object without waiting for another device to load it. This means that objects can be temporarily placed at the transfer point before being loaded onto another device. Some examples are temporary storage places where tugger trains or forklifts unload/load volumes to be transferred between them.

A 'Meeting point' refers to a workstation where two devices synchronize their activities to transfer an object directly between them. This means that at the same time, the object is unloaded from one device and loaded onto the other. For this to happen smoothly, both devices must be located at the same workstation. If they are not, they may have to wait for each other. Some examples are the transfer between an AGV and an automated robot arm that places items in the AGV, or between a crane and a truck.

Graphic Representation: The representation of a workstation, a transfer point and a meeting point is based on the symbols presented in Figure 3. For transfer and meeting points, the number of circles must be proportional to the number of devices that interact with the

workstation. Tables are optional graphic elements proposed to show additional information of each element—in this case, to present the physical location of each workstation.

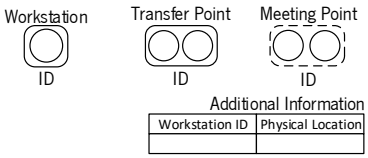


Figure 3. Graphic representation of a workstation.

Example: Table 2 shows some examples of the definition of workstations. Workstations WK1 and WK2 can represent an execution workstation, a warehouse or an external entity. TP1 can represent, for instance, a place where tugger trains unload and load items to be transferred between them. MP1 can represent, for instance, a workstation where the transfer of an object occurs between an AGV and an automated robot arm that places items in the AGV, or between a crane and a truck.

Table 2. Example of the definition of workstations.

Workstation ID	Workstation Type		Physical Location
WK1	Workstation	WK1	WK1
WK2	Workstation	WK2	WK2
TP1	Transfer Point	TP1	TP1
MP1	Meeting Point	MP1	MP1

Apart from the listed properties, each workstation can also be organized in various physical areas to locate objects.

3.2.1. Storage Location

A **storage location** is an object that can be transported by a device (as referred to in the previous section) and also a physical area within a workstation where volumes can be placed, and that may (or may not) be transportable by a device. Each storage location (SL) is characterized by the following properties [19]:

- Origin workstation [SL_{ORIGN_WK}]: workstation WK_{ID} of the storage location;
- Storage location ID [SL_{ID}]: unique identifier of the storage location;
- Transportable [$SL_{TRANSPORTABLE}$]: property to define if the storage location can be transportable by a device to other workstations. When a storage location can be transported by a device, it will always be a storage location that belongs to the origin workstation SL_{ORIGN_WK}, but it can be transported to other workstations in order to be unloaded or to allow the picking/storing of volumes at other workstations. The value domain of this property is {'Yes', 'No'};
- Physical location [$SL_{LOCATION}$]: physical location of the storage location.

Example: Table 3 shows some examples of the definition of two storage locations—SL1 and SL2. Storage locations have no graphic representation, but when required, these elements and the remaining ones presented in this section can be represented in a table—like in Figure 3.

Table 3. Example of the definition of a storage location.

Origin Workstation	Storage Location ID	Transportable	Physical Location
WK1	SL1	Yes	#01
WK1	SL2	No	#02

3.2.2. Storage Location Position

A storage location can have multiple physical areas for placing volumes, which is important for controlling handling activities such as picking and storing at workstations.

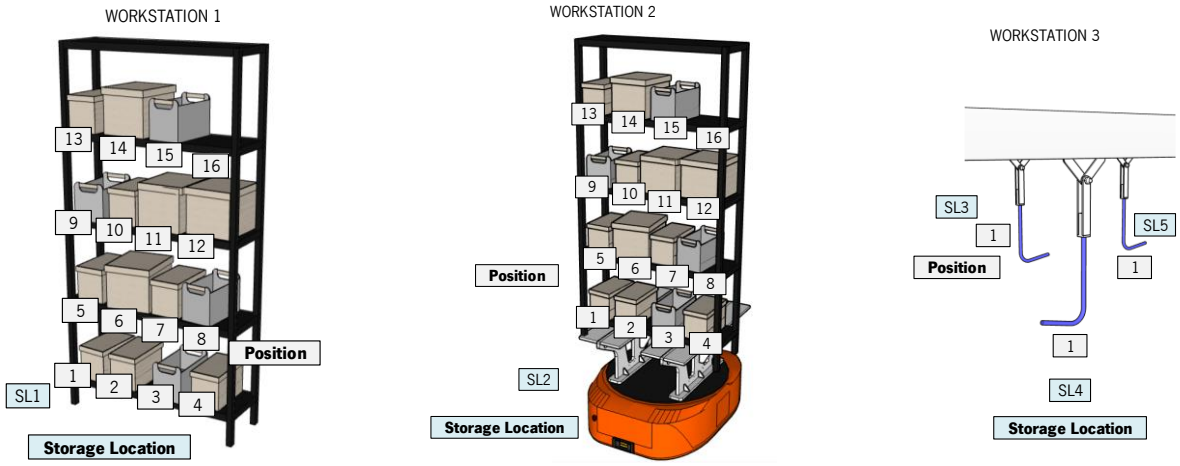
A **storage location position** is a specific area within a workstation's storage location where only one volume can be stored. A storage location can have a set of positions, and each storage location should have at least one position—otherwise no volumes can be placed at the storage location. A storage location and a storage location position represent two hierarchical levels to identify the physical area where a volume is placed in a workstation. The concept of storage location position is introduced in the representation language because a storage location of a workstation can also be transported to other workstations in order to pick or store the required volumes—under these circumstances, the storage location position allows the definition of where a volume is placed or where it should be stored in a storage location. A position is not considered an object to be transported.

Each position of a storage location (PTSL) is characterized by the following properties [19]:

- Storage location [$PTSL_{SL}$]: storage location SL_{ID} where the position is associated;
- Storage location position ID [$PTSL_{ID}$]: unique identifier of the storage location position;
- Restriction [$PTSL_{RESTRICTION}$]: mathematical expression to define if a volume can or cannot be stored at storage location position $PTSL_{ID}$. The definition of a restriction implies different input data and mathematical expressions to evaluate if a volume can or cannot be placed at a storage location position—therefore, restrictions are defined by each organization. The operands of the mathematical expression correspond to the volume parameters (VPs) defined in the system. For instance, if the restriction is VP1='Raw Material', it means that what can only be stored at this storage location position are volumes whose volume parameter VP1 satisfies this condition;
- Physical location [$PTSL_{LOCATION}$]: physical location of the position.

Example: Figure 4 shows three examples to show the comprehensiveness of the properties proposed to represent how each workstation is organized:

- Figure 4a illustrates a storage structure that cannot be transported. Workstation 1 is characterized by one storage location (SL1) with several positions (1 to 16);
- Figure 4b illustrates a storage structure that can be transported by a device. Workstation 2 has also one storage location (SL2) with several positions (1 to 16). The main difference is that the storage location identifies a storage structure that can be transported to different workstations (storage location SL2). Positions identify where each volume is placed in the storage structure in order to be picked or stored volumes;
- Figure 4c illustrates a storage structure organized by hooks. Workstation 3 is defined by three storage locations (SL3 to SL5), and each storage location has only one position. Each hook is represented as a storage location position—so, each hook can store a volume, and whenever required, the right hook should be moved to a particular point.



(a) **(b)** **(c)**

Figure 4. Examples of workstations with different characteristics: (**a**) Traditional storage structure; (**b**) Transportable storage structure; (**c**) Storage structure organized by hooks.

Table 4 shows some examples of the definition of positions for storage location SL1. At the storage location position PTSL2, it can only be stored a volume whose volume parameter VP2 (weight—in kg) is lower than 100 kg.

Table 4. Example of the definition of storage location parameters and positions.

Storage Location	Storage Location Position ID	Restriction	Physical Location
SL1	PTSL1	VP1 = 'Raw Material'	#001
SL1	PTSL2	VP2 < 100	#002

3.2.3. Storage Location Parameter Value

Each storage location is also characterized by the storage location parameters (SLPs) defined by the organization in the system. Each parameter value of a storage location (SLPV) is characterized by the following properties [19]:

- Storage location [SLPV $_{VL}$]: storage location SL_{ID};
- Storage location parameter [SLPV $_{SLP}$]: storage location parameter SLP_{ID};
- Value [$SLPV_{VALUE}$]: value of the parameter—based on the domain defined in the storage location parameter property SLP_{DOMAIN}.

Example: Table 5 shows the definition of two storage location parameters for storage location SL1. SLP1 refers to the parameter that identifies the size of the storage location (small, medium or large). SLP2 refers to the height (in cm) of the storage location.

Table 5. Example of the definition of storage location parameters.

Storage Location	Storage Location Parameter	Value
SL1	SLP1	Medium Storage Rack
SL1	SLP2	100

3.3. Device

A TMHS consists of one or more devices. A **device** is a transport and handling equipment of the TMHS of a particular organization. Each device (DV) is defined by the following properties [19]:

- Device ID [DV_{ID}]: unique identifier of the device;
- Object type [DV_{OBJECT}]: object type that can be transported by the device. The value domain of this property is {'Volume', 'Storage Location'}.

Object type [DV_{OBJECT}]: The main reason to distinguish volumes and storage locations are the activities that should be executed at each workstation:

- A volume must be picked at a workstation, loaded into the device and transported to another workstation, and the volume should be unloaded there;
- A storage location should be loaded at a workstation and transported to another workstation where one or more volumes must be picked or stored—therefore, the picking and storing only occur when the device that transports the storage location arrives at the workstation. In order to pick and store each volume, the storage location can be unloaded at a workstation or the device can wait at the workstation until these activities are executed. Then, the storage location has to be moved to another workstation or returned to the origin workstation SL_{ORIGN_WK}.

  Example: Table 6 shows an example of the definition of two devices—DV1 and DV2.

Table 6. Example of the definition of two devices.

Device ID	Object Type
DV1	Volume
DV2	Storage Location

3.3.1. Device Position

Like a workstation, a device has physical areas where objects can be placed. A **device position** is a place of the device where only one object can be stored—either a volume or a storage location. A device can have several device positions, and each device should have at least one device position—otherwise, no objects can be loaded at the device. Each position of a device (PTDV) is defined by the following properties [19]:

- Device [$PTDV_{WK}$]: device DV_{ID} where the position is associated;
- Device position ID [$PTDV_{ID}$]: unique identifier of the device position;
- Restriction [$PTDV_{RESTRICTION}$]: mathematical expression to validate if an object can be placed in the position. The operands are defined based on the volume parameters (VPs) and storage location parameters (SLPs) defined in the system;
- Physical location [$PTDV_{LOCATION}$]: physical location of the position.

The next sections present the set of properties to characterize the behavior of each device and to represent the behavior of each activity when a particular device executes it.

3.3.2. Device Route Link

A device moves between workstations to handle and transport objects based on the manufacturing system and internal procedures. The sequence of workstations visited can take various shapes, such as circular or a star [22].

A **device route link** represents a link between two workstations that can be executed by a device. Each device has a set of device route links that identify between which workstations the device can move. When there are device route links with the same origin

workstation, the processes that manage transport activities can identify which device route link should trigger an activity to move to another workstation.

Figure 5 illustrates a TMHS with three devices. Device 2 can visit six workstations and has seven device route links. Two of the device route links (1 and 2) have the same origin workstation WK1, which means that, when device 2 is at WK1, the device can move to WK2 or WK3. An important point that must be considered by users in the definition of device route links for a particular device is that all workstations that are destinations should also be an origin workstation of a different device request link—otherwise, there is no information about where the device can go when the device is located at the workstation, and the device will stop.

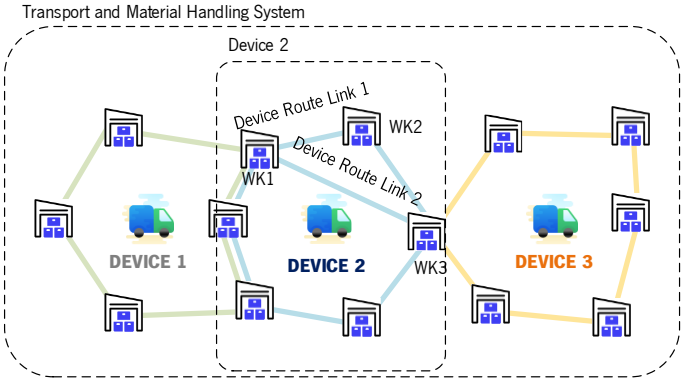


Figure 5. Example of a TMHS with three devices.

Each device route link (DVRL) is defined by the following properties [19]:

- Device [$DVRL_{DV}$]: device DV_{ID} where the device route link is associated;
- Device route link ID [$DVRL_{ID}$]: sequential identifier of each device route link of the device $DVRL_{DV}$. The identifier should start at 1 for each device and increase;
- Origin workstation [$DVRL_{ORIGIN_WK}$]: origin workstation WK_{ID};
- Destination workstation [$DVRL_{DESTINATION_WK}$]: destination workstation WK_{ID};
- Time [$DVRL_{TIME}$]: time to move from the origin workstation $DVRL_{ORIGIN_WK}$ to the destination workstation $DVRL_{DESTINATION_WK}$;
- Execution entity [$DVRL_{EXECUTION}$]: property to define if the activity should be executed by the device or automatically—when there is no need to control its execution. The value domain of this property is {'Device', 'Automatic'};
- Go-To template [$DVRL_{TEMPLATE}$]: template to trigger an instruction for a device to move between two workstations. An instruction is dispatched every time the movement of the device is triggered based on the device route link $DVRL_{ID}$. A go-to template considers one template parameter (activity ID) that represents an identifier of the activity triggered;
- Restriction [$DVRL_{RESTRICTION}$]: mathematical expression to define if the device route link can be used to transport an object. The operands are defined based on the volume parameters (VPs) and storage location parameters (SLPs) defined in the system. Some examples are the product type (raw materials, semi-finished, work-in-process, final goods, dangerous materials, etc.), the material application (manufacturing, maintenance, setup, etc.) or the shape of each object (box, reel, container, pallet, etc.) that can be transported in a device.

Go-To template [$DVRL_{TEMPLATE}$]: Some industrial equipment and systems require specific instructions to execute tasks within a manufacturing system and ensure interoperability between systems. Each instruction is unique due to the variety of equipment functions, technologies and suppliers involved. For example, instructions for moving an AGV differ

from those for automated storage equipment, and even within the same equipment type, instructions can vary based on the supplier, destination workstation or specific task.
The European Interoperability Framework identified three layers of interoperability: syntactic, semantic and organizational. Syntactic interoperability deals with the technical challenges of connecting systems at the lowest level. Service-oriented architectures (SOAs), including SOAP, REST and JSON, are widely recognized in the industry, and organizations intend to leverage the existing ones instead of developing new ones. Semantic interoperability establishes guidelines for understanding the meaning of data, allowing it to be processed within one system before being transferred to another. Organizational interoperability focuses on aligning organizational structures and business processes [23,24].

A **template** is, in the representation language, a form that defines the data structure for triggering instructions sent to equipment or a system—ensuring semantic interoperability with industrial equipment. It consists of a sequence of characters identifying a specific function of the equipment, along with template parameters. The **template parameters** are variables whose values must be assigned based on the activity to be executed, creating specific instructions with specific data. An instruction to trigger the movement of a device in the manufacturing system is represented as a **go-to template**.

Some examples of equipment that can require go-to templates are an AGV, an AGV goods-to-person or a self-driving vehicle. Figure 6 shows a schema of the definition of a Go-To template. The structure applied in the template to communicate was collected from an AGV—where instructions must be sent to the AGV in order to move inside the shop floor. The structure of the instruction is related to the type of protocol to communicate with the equipment—in this case, the equipment receives instructions based on SOAP web services. It is important to note that, although this example uses SOAP web services to communicate, the same principle would be followed with different approaches. Other equipment may require different types of protocols or formats to communicate (such as REST or JSON), and the template concept would be applied in the same way. In the future, the application of the template concept with meta-models for Industry 4.0 digital twins, such as Asset Administration Shell, should also be explored.

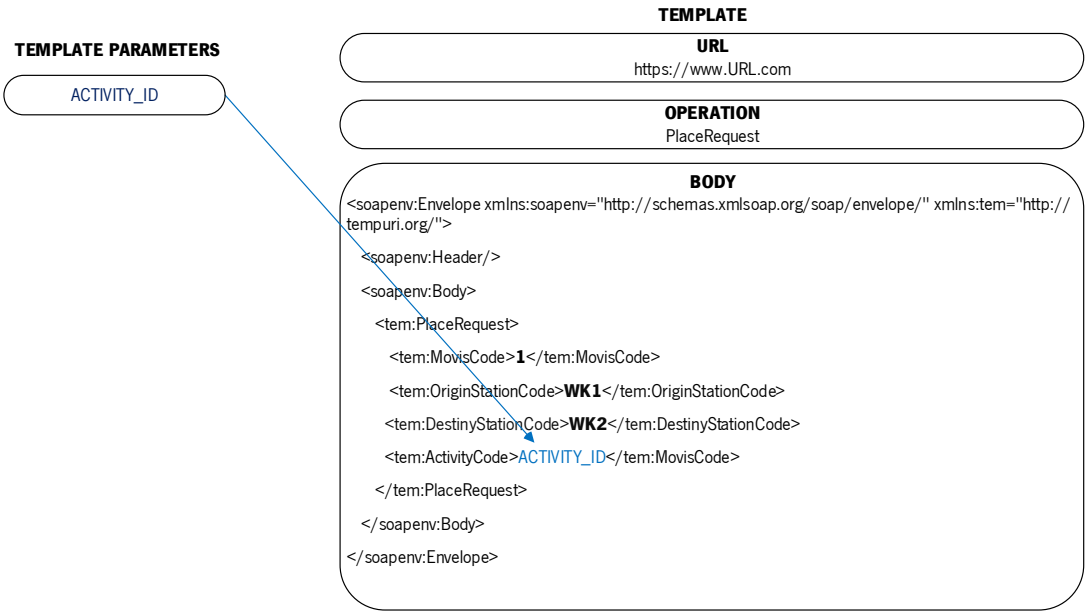


Figure 6. Example of the definition of a go-to template.

The SOAP web service is used as a case study to show how the template concept can be applied in Figure 6. A SOAP web service requires the definition of the URL, operation

(function of the web service) and body of the request. The body of the request must contain all needed information according to the structure of the web service function. This way, the definition of the Go-To template shown in Figure 6 requires the following:

- The definition of the URL, the operation, the AGV (referred to as "MovisCode" in the body), the origin workstation ("OriginStationCode"), the destination workstation ("DestinationStationCode") and the remaining structure of the instruction—all these data refer to a sequence of characters that should be identified directly by the organization in the template;
- The definition of template parameter ACTIVITY_ID (represented in blue)—this parameter should be automatically assigned by the processes that manage/control transport activities according to the activity that the device should execute.

Graphic Representation: Figure 7a shows a graphic representation of a device route link—from the origin workstation to the destination workstation (the arrow indicates the direction). Each workstation is represented according to the workstation type (Figure 3). Figure 7b shows the graphic representation of both object types that can be transported by a device—continuous lines refer to the transport of volumes, and dashed lines refer to the transport of storage locations. Each set of device route links defined for a particular device should be represented with the same color.

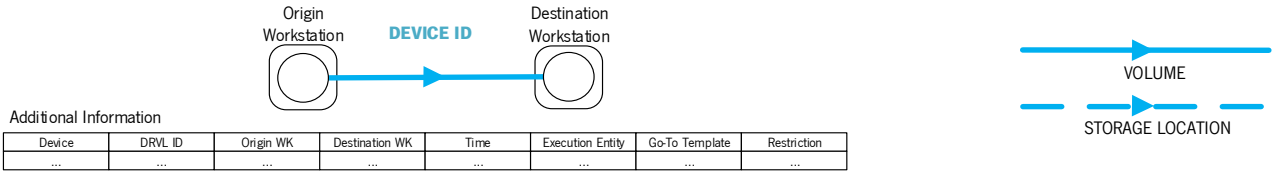


Figure 7. Graphic representation of (**a**) Device request link; (**b**) Device request link of each object type that can be transported.

Figure 8 represents the device request links of a TMHS with three devices—devices DV1, DV2 and DV3. Device route links of different devices are represented with different colors.

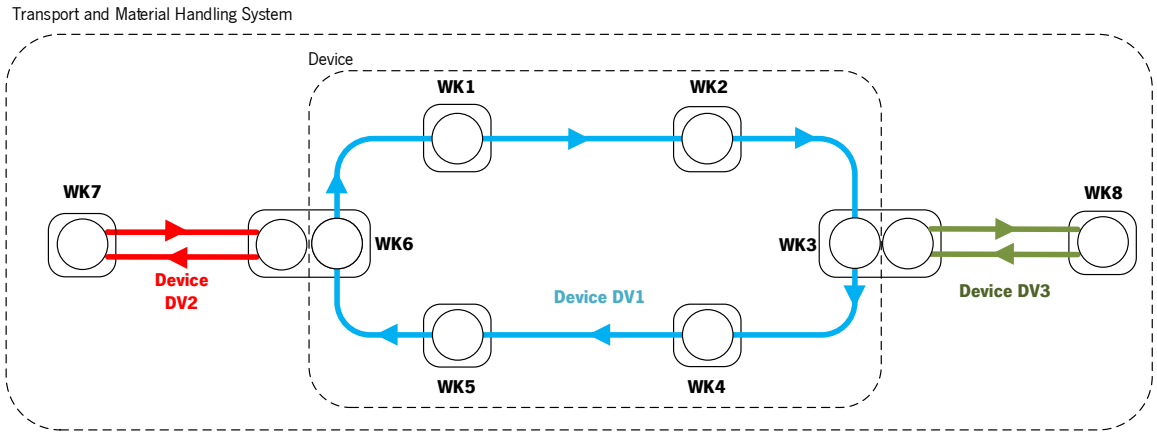


Figure 8. Representation of the device route links for a TMHS that consists of three devices (DV1, DV2 and DV3).

Example: Table 7 shows the definition of the device route links of the TMHS represented in Figure 8. Device DV1 can only transport raw materials. Device DV2 has go-to templates to trigger instructions and can only transport volumes that weigh less than 100 kg.

Table 7. Example of the definition of device route links.

Device	Device Route Link ID	Origin Workstation	Destination Workstation	Time	Execution Entity	Go-To Template	Restriction
DV1	1	WK1	WK2	5	Device		VP1 = 'Raw Material'
DV1	2	WK2	WK3	5	Device		VP1 = 'Raw Material'
DV1	3	WK3	WK4	5	Device		VP1 = 'Raw Material'
DV1	4	WK4	WK5	5	Device		VP1 = 'Raw Material'
DV1	5	WK5	WK6	5	Device		VP1 = 'Raw Material'
DV1	6	WK6	WK1	5	Device		VP1 = 'Raw Material'
DV2	1	WK6	WK7	2	Device	Template A	VP2 $\leq$ 100
DV2	2	WK7	WK6	3	Device	Template B	VP2 $\leq$ 100
DV3	1	WK3	WK8	5	Device		
DV3	2	WK8	WK3	5	Device		

3.3.3. Device Route Activity

A **device route activity** represents a handling activity that can be executed at the destination workstation of a device route link (property $DVRL_{DESTINATION_WK}$). This way, each device is characterized by a route that consists of an ordered set of device route links and activities that a device can execute. It was defined that each device has a specific route (instead of having routes that could be assigned to different devices) because the activities that each device can execute depend in part on the characteristics of the device.

Device route activities are defined for each destination workstation $DVRL_{DESTINATION_WK}$, and not directly to a workstation, to allow the representation of different approaches to transporting objects. This feature of the language gives users the flexibility to define handling activities to be executed depending on the origin workstation from where the device departs. For example, a device may be able to load and unload volumes at a workstation during the same visit, or the device may need to deliver all volumes to all workstations first and then load new volumes [25].

Each device route activity (DVRA) is defined by the following properties [19]:

- Device [$DVRA_{DV}$]: device DV_{ID} where the activity is associated;
- Device route link [$DVRA_{DVRL}$]: device route link $DVRL_{ID}$ where the activity is associated;
- Device route activity ID [$DVRA_{ID}$]: sequential identifier of each activity of the device route link $DVRA_{DVRL}$;
- Action type [$DVRA_{ACTION}$]: property to define what can be executed at the destination workstation of the device route link $DVRA_{DVRL}$. The value domain depends on the object type property of device $DVRL_{DV}$. If the object type is 'Volume', the value domain is {'Pick and Load', 'Unload and Store'}. If the object type is 'Storage Location', the value domain is {'Load', 'Unload', 'Pick' and 'Store'};
- Synchronization mode [$DVRA_{SYNCHONIZATION}$]: property to define when a volume can be unloaded after the device arrives at a workstation. The value domain depends on the object type property of device $DVRL_{DV}$, action type $DVRA_{ACTION}$ and workstation type WK_{TYPE}—if the object type is 'Volume', the action type is 'Unload and Store' and the workstation type is 'Workstation', the value domain is {'On arrival', 'On execution', 'On finishing', 'Never'}. For the remaining options, this property can only be defined as 'On arrival';
- Execution entity [$DVRA_{EXECUTION}$]: property to define who must execute the activity. The value domain of this property is {'Device', 'Workstation', 'Automatic'};

- Input required [$DVRA_{INPUT}$]: property to define if an external input is required to confirm the execution of the activity—in order to trigger a new activity to the device. The value domain of this property is {'Yes', 'No'};
- Handling template [$DVRLA_{TEMPLATE}$]: template to trigger an instruction. An instruction is triggered by each object handled—that is, one instruction per volume or one per storage location. The template parameters associated with a handling template are activity ID, volume, storage location, storage location position and device position. Each template parameter identifies information to define from where one object should be picked or where it should be placed—according to the action type $DVRA_{ACTION}$;
- Restriction [$DVRLA_{RESTRICTION}$]: mathematical expression to validate if the device route activity can be executed by the device. The operands are defined based on the volume parameters (VPs) and storage location parameters (SLPs) defined in the system. Some examples of restrictions that can be defined in loading activities are the maximum weight, space, number of volumes or storage locations that can be loaded into the device.

Action type [$DVRA_{ACTION}$]: The identification of the action types that each device can execute at each device route link is important to represent internal organizational procedures, characteristics of the equipment and features of the manufacturing system. A device, such as a tugger train, a forklift or a hand pallet truck, typically can execute several action types such as loading or unloading, but the same device may be only authorized to load volumes at specific workstations. Different terminologies are applied in the literature to identify different action types, such as pick up, load, unload, go-to, storage, delivery, put away, arrive, etc. [7,8,26]. The following action types have been identified in the representation language: If the object type transported by the device DV_{OBJECT} is a volume:

- Pick and load: the action of selecting a volume and placing it in a device position. The volume can be loaded from a storage location position or a device position (for meeting points, volumes are picked from device positions);
- Unload and store: the action of taking out a volume of a device position and placing it at a storage location position or device position (meeting points).

If the object type transported by the device DV_{OBJECT} is a storage location:

- Load: the action of selecting a store location and placing it at a device position. The storage location can be loaded from a workstation or a device position (meeting points);
- Unload: the action of taking out a storage location of a device and placing it at a workstation or device position (meeting points);
- Pick: the action of selecting a volume from a storage location position that is placed in a device;
- Store: the action of placing a volume in a storage location position that is placed in a device.

The identification of the action types that each device can execute at each device route link is important to represent internal organizational procedures, characteristics of the equipment and features of the manufacturing system. A device, such as a tugger train, a forklift or hand pallet truck, typically can execute several action types such as loading or unloading, but the same device may be only authorized to load volumes at specific workstations.

Synchronization mode [$DVRA_{SYNCHONIZATION}$]: Each synchronization mode requires different information to define the trigger that requests a specific transport in the manufacturing system—referred to in this paper as a job. A job can be any operation that triggers the transport of a volume, such as an operation where a volume may be incorporated or an operation that may be executed at the volume. For instance, a volume can refer to a set of

parts that need to be transported to be incorporated in a particular manufacturing batch (job). Or a volume can refer to a set of products that are being manufactured and that needs to be transported to a workstation in order to execute an operation (job) on the products.

The synchronization modes identified to characterize a device route activity are as follows:

- On arrival: represents the behavior when volumes are unloaded right after the device arrives at the workstation;
- On execution: represents the behavior when volumes are unloaded at the workstation only after the device arrives at the workstation and the job that requires the volume starts its execution. Even if the device is located in the workstation, the device should wait until the job starts. After that, the volume can be unloaded. An example is when a volume can only be unloaded in the execution area of an execution workstation;
- On finishing: represents the behavior when volumes are unloaded at the workstation only after the device arrives at the workstation and the job that requires the volume finishes its execution, for instance, when items are consumed directly from the volume that is placed in the device;
- Never: represents the behavior when a device has to stop at the workstation (because a volume is needed there), but the volume is not unloaded at the workstation—the volume should remain loaded in the device in order to be transported to a new workstation. The device can only go to the next workstation when the job that triggered the transport of the volume finishes its execution. An example is an AGV Assembly Line because the AGV transports a volume throughout several workstations to execute jobs without unloading the volume (for instance, a car body).

The transport of a storage location is the transport of several volumes whose transport may have been triggered by different jobs—so, the synchronization mode must be always 'On arrival'.

Execution entity [$DVRA_{EXECUTION}$]: The execution entities identified to define a device route activity are as follows:

- Device: represents the behavior when the device must execute the activity;
- Workstation: represents the behavior when the workstation must execute the activity—the workstation where the device is located;
- Automatic: represents the behavior when the activity triggered must be executed automatically. This behavior should be applied when it is important to identify the activity but there is no need to control if the activity has been effectively executed—for instance, to trigger an instruction to an equipment but the equipment is not able to confirm the execution of the activity.

The identification of the execution entity depends on the characteristics and functions of the equipment—as illustrated in Figure 9. For instance, an AGV equipped with a powered roller conveyor or an automated robotic arm can load a volume automatically and autonomously (Figure 9a)—so, the execution entity is the device. An AGV without handling equipment requires another resource to load volumes. The resource can be an external piece of handling equipment or a human operator (Figure 9b)—so, the execution entity is the workstation.

In other cases, the execution of these activities may be controlled by the fleet management system of the equipment. A fleet management system is responsible for defining which tasks should be executed by each piece of equipment and the management of all processes from the start to finish of each transport (load capacities, vehicle collision, traffic congestion, equipment speed, departure times, etc.) [27].



(a) **(b)**

Figure 9. Examples of AGVs: (**a**) AGV with handling equipment (arm robot); (**b**) AGV without handling equipment.

For instance, an AGV fleet management platform should receive information about the activities that should be executed—but the definition of the equipment to execute each activity and the monitoring of its execution is ensured by its own management system. The same principle can be followed by overhead logistic conveyor systems to identify which hooks must be moved to execute each activity. Under these circumstances, communication may be required to identify what needs to be executed, but the activities are managed by the equipment management system—so, the execution entity should be automatic.

Input required [$DVRA_{INPUT}$]: Some devices, such as a tugger train, a forklift or an AGV, can only be located at one workstation and execute one device route activity at each time. Other devices can execute activities simultaneously at different workstations. For instance, a conveyor can load or unload several volumes at different workstations at the same time. Other examples, such as AGV fleet management platforms or overhead logistic conveyor systems, have similar behavior because they do not refer to a single equipment but a group of equipment—so, they can also execute several activities simultaneously at different workstations. Transport system services, such as transport by train or truck, typically also present this behavior.

This way, two alternatives have been identified to define the input required property:

- Yes: represents the behavior when the device route activity should be executed before a new activity to the device can be triggered. The execution of the activity depends on an external event (input) to confirm that the activity was executed;
- No: represents the behavior when a new activity can be triggered without the confirmation of the conclusion of the activity that is being defined.

Handling template [$DVRLA_{TEMPLATE}$]: The template parameters available for handling templates are important because different equipment may require different parameters. Some examples of information required are:

- Volume to handle: some equipment require the identification of the right volume to be handled because equipment have their management systems;
- Storage location position: some equipment require the identification of the storage location position to handle. For instance, an instruction can be sent to an automated storage and retrieval equipment based on the identification of the storage location position to activate the robotic arm and transfer a box to an unloading point. The overhead logistic conveyor system can also require the storage location position to identify which hook should be activated and moved to an unloading point.

A scheme of the definition of a handling template to pick and load a volume is presented in Figure 10—similar to the definition of a go-to template (Figure 6). The main difference refers to the information defined in the template and the template parameters available.

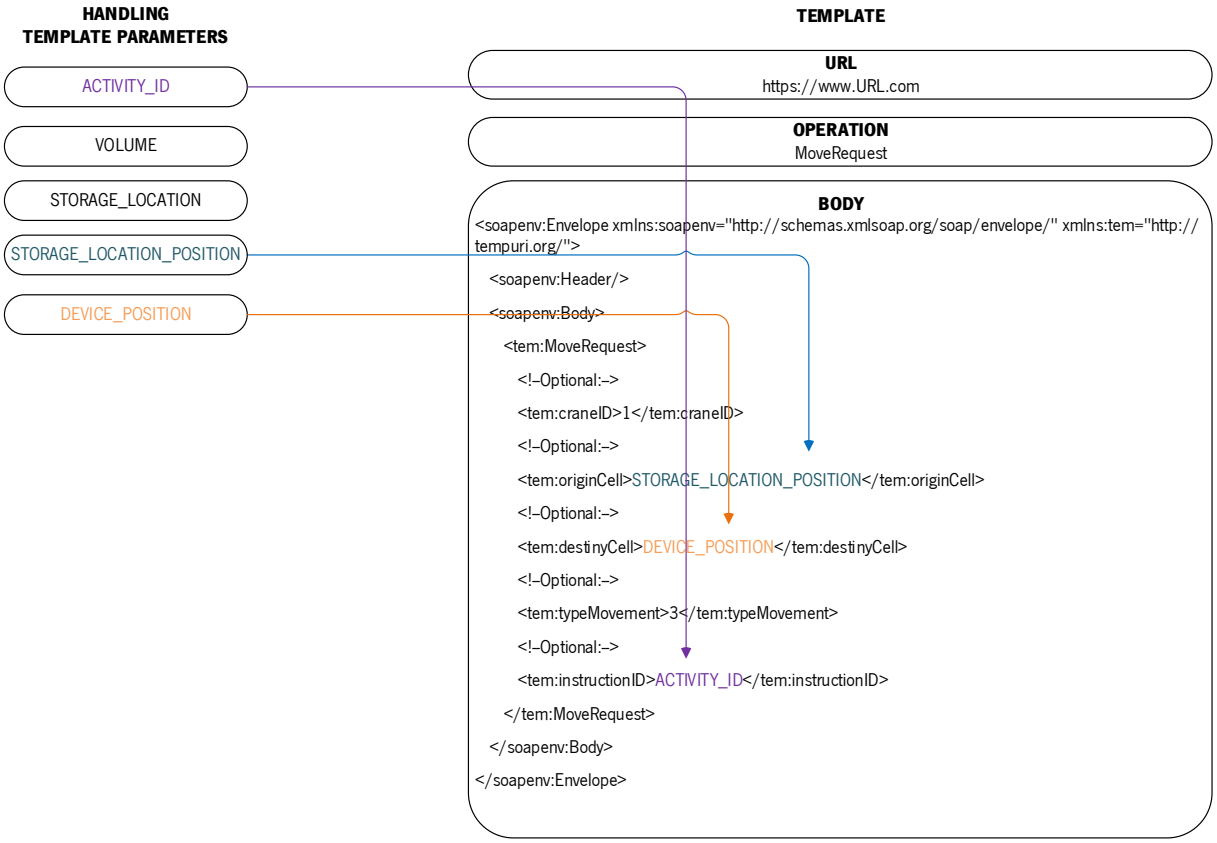


Figure 10. Example of the definition of a handling template.

The structure of the template shown in Figure 10 was collected from one automated storage and retrieval equipment—where instructions must be sent to handle each volume. The definition of the template requires three template parameters:

- Storage location position: storage location position from where the volume should be picked;
- Device position: device position where the volume should be placed;
- Activity ID: identifies the activity that triggered the instruction. After the execution of the activity, the equipment can use this parameter to identify which activity was executed and send that information back to the processes that are controlling transport and handling activities.

Graphic Representation: Figure 11 shows a graphic representation of device route activities based on the action type, and a device route that consists of six device route links between six workstations (WK1 to WK6) and a set of device route activities that can be executed at each workstation. Device 1 can load and unload volumes at all workstations during the same visit. The sequence of the device route activities depicted at each link represents the sequence defined by the activity ID property $DVRA_{ID}$. The representation of the device route also includes two tables with additional information on each device route link and activity.

Figure 12 represents a complete TMHS with three device routes—devices DV1, DV2 and DV3.

Example: Table 8 shows the definition of the device route activities of the TMHS represented in Figure 12—considering the device route links of Table 7. Device DV2 can only pick and load volumes at workstation WK07 (but only volumes that weigh less than 100 kg). The restriction of device DV3 identifies the maximum weight that can be loaded at the device

(since device DV3 can only pick and load volumes at workstation WK3 until the sum of all volumes loaded is less than 500 kg).

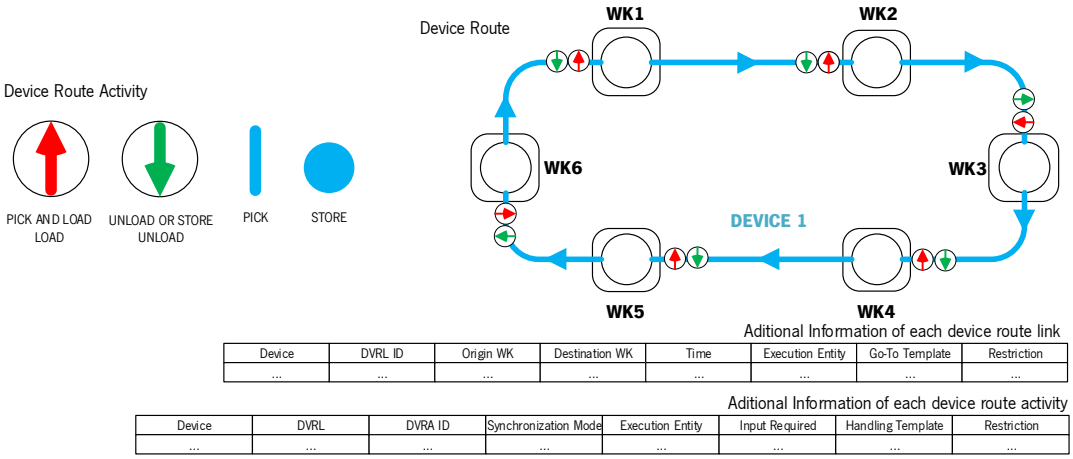


Figure 11. Graphic representation of a device route.

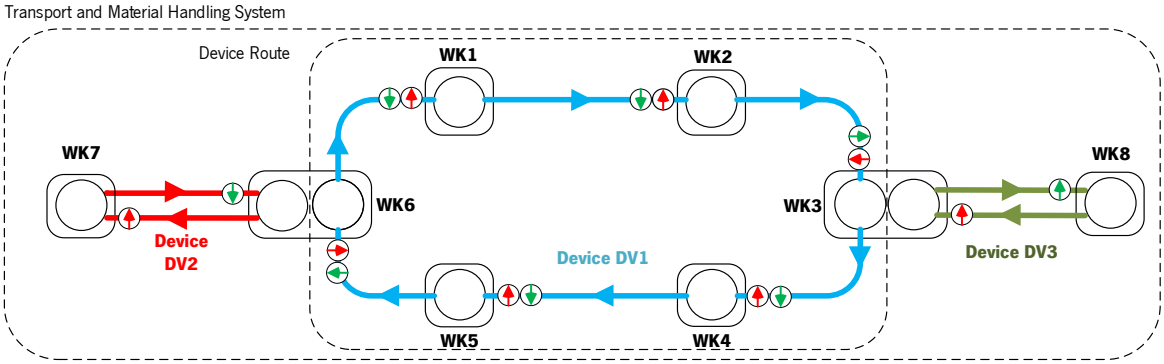


Figure 12. Representation of the device route links and activities for a TMHS that consists of three devices (DV1, DV2 and DV3).

Table 8. Example of the definition of device route activities.

Device	Device Route Link	Device Route Activity ID	Action Type	Synchronization Mode	Input Required	Handling Template	Restriction
DV1	1	1	Unload and Store	On arrival	Yes		
DV1	1	2	Pick and Load	On arrival	Yes		
DV1	2	1	Unload and Store	On arrival	Yes		
DV1	2	2	Pick and Load	On arrival	Yes		
DV1	3	1	Unload and Store	On arrival	Yes		
DV1	3	2	Pick and Load	On arrival	Yes		
DV1	4	1	Unload and Store	On arrival	Yes		
DV1	4	2	Pick and Load	On arrival	Yes		
DV1	5	1	Unload and Store	On arrival	Yes		
DV1	5	2	Pick and Load	On arrival	Yes		

Table 8. *Cont.*

Device	Device Route Link	Device Route Activity ID	Action Type	Synchronization Mode	Input Required	Handling Template	Restriction
DV1	6	1	Unload and Store	On arrival	Yes		
DV1	6	2	Pick and Load	On arrival	Yes		
DV2	1	1	Pick and Load	On arrival	Yes	Template C	VP2 $\leq$ 100
DV2	2	1	Unload and Store	On arrival	Yes	Template D	
DV3	1	1	Unload and Store	On arrival	Yes		
DV3	2	1	Pick and Load	On arrival	Yes		Sum(VP2) $\leq$ 500

4. Results

The results of the application of the generic representation language developed depend on its ability, flexibility and comprehensiveness to represent physical and behavioral characteristics of TMHS. The approach followed to validate the language was to represent the highest number of selected devices and TMHS characterized by different behaviors, rules and restrictions.

This section is divided into two subsections. The first subsection shows the application of the representation language to represent devices characterized by different behaviors and relates each representation to examples of real and industrial transport and handling equipment. The second subsection shows the representation of three case studies of TMHS presented in the literature with the language presented. The objective is to compare the information presented in each case study with the information represented by the language.

4.1. Representation of TMHS

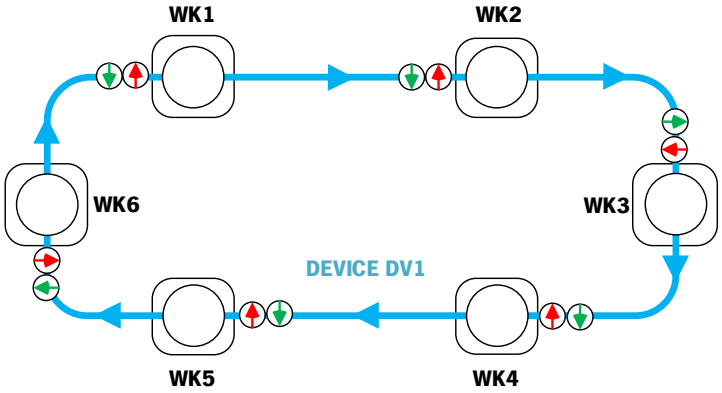
This section starts with a representation of a device route using the language proposed, and then the representation is matched with real examples of industrial transport and handling equipment.

Figure 13 shows the representation of a device DV1 with the following characteristics:

- The device can visit six workstations (WK1 to WK6);
- The device can 'Pick and load' and 'Unload and store' volumes at all workstations;
- Volumes can be unloaded after the device arrives at each workstation—the synchronization mode $DVR_{SYNCHRONIZATION}$ is 'On arrival' for all device route activities;
- The device can only execute one device activity at each time—the input property $DVRL_{INPUT}$ is defined as 'Yes' for all device route activities;
- All device activities should be executed by the device—the execution entities $DVRL_{EXECUTION}$ and $DVRA_{EXECUTION}$ are defined as 'Device' for all device route links and activities;
- All device activities should trigger an instruction based on the go-to and handling templates.

The representation of the device shown in Figure 13 could be used to represent an AGV with handling equipment (such as a powered roller conveyor or a robotic arm), where each workstation represents a tag where the AGV can go to load or unload volumes—mainly because the device can only move over a rigid sequence of device route links (there are no device route links with the same origin workstation), go-to and handling templates are defined to send instructions to the equipment, and the object type is volume. Moreover, the device route can be used to represent the transport of different volumes

(such as raw materials or manufacturing batches) between different workstations (such as warehouses or execution workstations)—which can have different physical areas where volumes are placed.



Aditional Information of each device route link

Device	DVRL ID	Origin WK	Destination WK	Time	Execution Entity	Go-To Template	Restriction
DV1	1	WK1	WK2	5	Device	Template A	
DV1	2	WK2	WK3	5	Device	Template B	
DV1	3	WK3	WK4	5	Device	Template C	
DV1	4	WK4	WK5	5	Device	Template D	
DV1	5	WK5	WK6	5	Device	Template E	
DV1	6	WK6	WK1	5	Device	Template F	

Aditional Information of each device route activity

Device	DVRL	DVRA ID	Action Type	Synch. Mode	Execution Entity	Input Required	Handling Template	Restriction
DV1	1	1	Unload and Store	On Arrival	Device	Yes	Template G	
DV1	1	2	Pick and Load	On Arrival	Device	Yes	Template H	
DV1	2	1	Unload and Store	On Arrival	Device	Yes	Template G	
DV1	2	2	Pick and Load	On Arrival	Device	Yes	Template H	
DV1	3	1	Unload and Store	On Arrival	Device	Yes	Template G	
DV1	3	2	Pick and Load	On Arrival	Device	Yes	Template H	
DV1	4	1	Unload and Store	On Arrival	Device	Yes	Template G	
DV1	4	2	Pick and Load	On Arrival	Device	Yes	Template H	
DV1	5	1	Unload and Store	On Arrival	Device	Yes	Template G	
DV1	5	2	Pick and Load	On Arrival	Device	Yes	Template H	
DV1	6	1	Unload and Store	On Arrival	Device	Yes	Template G	
DV1	6	2	Pick and Load	On Arrival	Device	Yes	Template H	

Figure 13. Representation of device DV1.

On the other hand, this representation can also be used for single integrated equipment (such as an overhead logistic conveyor system or automated storage and retrieval equipment) because, although they refer to a single piece of equipment of the manufacturing system, analyzing with more detail, their behavior is similar to a device with several loading and unloading points across its route—WK1 could represent the storage area, and the remaining workstations could represent loading and unloading points of the equipment. For an overhead logistic conveyor system, each storage location position of the workstation WK1 would represent a hook where a volume is placed. For automated storage and retrieval equipment, each storage location position would represent a position of the storage area.

Table 9 presents some representations of device routes considering as a starting point the representation of Figure 13—the first column identifies which properties should change compared to Figure 13, and the second column identifies some examples. The objective is to show how similar representations can be applied to different devices considering different definitions of each property.

Table 9. Representation of different device route links and activities (considering as starting point the representation of Figure 13).

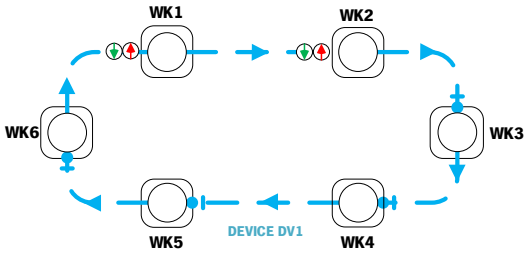
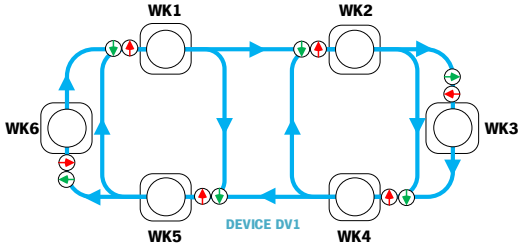
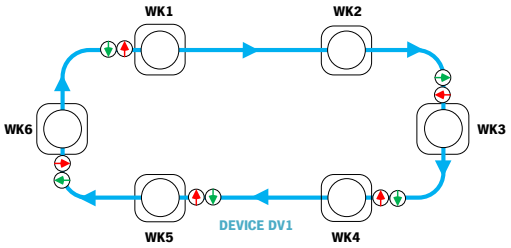
Representation (Main Differences from Figure 13)	Examples
Device—Object Type DV_{OBJECT} 	Devices that transport storage locations instead of the volumes. For instance, some AGV Goods to Person transport storage structures that are represented as storage locations. The action types of the device route activities executed at each workstation should also be redefined (as illustrated in the figure)—since the device may be able to load and/or unload storage locations at some workstations (for instance, origin workstations of the storage locations), and then volumes can be picked or stored at other workstations.
Device Route Link 	Devices that can depart from the same workstation to different workstations according to the activities that need to be executed. For instance, a self-driving vehicle or any transport equipment operated by an operator, such as a tugger train, a forklift, hand pallet trucks or other types of trucks. Equipment operated by an operator may not require instructions.
Device Route Link—Restriction $DVRL_{RESTRICTION}$ DVRL ID: —; **Restriction**: VP1='Raw-Material'	Devices that have restrictions to transport objects—due to equipment characteristics or internal organizational procedures. For instance, the restriction VP1 = 'Raw-Material' means that only volumes whose value volume parameter of VP1 is 'Raw-Material' can be transported by the device.
Device Route Activity 	Devices that have to move over a rigid sequence of device route links—even if device route activities are not executed at a workstation. In the figure, no device route activities are defined at workstation WK2, but the device DV1 is forced to move from WK1 to WK2. For instance, a device that has to visit WK6 to recharge the battery or a device that has to move over a fixed path, such as a magnetic tape.
Device Route Activity—Synchronization Mode $DVRA_{SYNCHONIZATION}$ DVRA ID: —; **Synchronization Mode**: On Execution / On Finishing	Devices that transport volumes that require synchronization with the job that triggered the transport request—namely, with the time that the job is started or finished. For instance, an AGV when volumes can only be unloaded in the execution area of an execution workstation or when items are consumed directly from the volume that is placed in the AGV.
Device Route Activity—Synchronization Mode $DVRA_{SYNCHONIZATION}$ DVRA ID: —; **Synchronization Mode**: Never	Devices that transport volumes throughout workstations without unloading the volume. For instance, a conveyor assembly line or an AGV car assembly line that carries a body car to be manufactured over workstations where jobs are executed—so, the volume is never unloaded from the device, and the device waits for the conclusion of the job at each workstation.
Device Route Activity—Execution Entity $DVRA_{EXECUTION}$ DVRA ID: —; **Execution Entity**: Workstation	Devices that transport the objects, but the handling device activities at the workstation are executed by the equipment—for instance, an AGV without handling equipment.

Table 9. *Cont.*

Representation (Main Differences from Figure 13)	Examples
Device Route Activity—Execution Entity $DVRA_{EXECUTION}$ DVRA ID: — / **Execution Entity**: Automatic	Devices that require only instructions to identify which objects need to be loaded or unloaded without controlling if these activities are executed. For instance, some overhead logistic conveyor systems require only one instruction for each volume that needs to be loaded and unloaded at each hook of the storage area—but the execution of these activities is controlled by the management system of the equipment.
Device Route Link—Execution Entity $DVRL_{EXECUTION}$ DVRL ID: — / **Execution Entity**: Automatic Device Route Activity—Input Required $DVRA_{INPUT}$ DVRA ID: — / **Restriction**: No	Devices that can execute different activities at different workstations at the same time (for instance, an AGV Fleet management or a conveyor can load and unload different volumes at different workstations at the same time). This way, there will be no control over the execution of the go-to device activities (because the device can be placed at any workstation at the same time), and several device activities can be triggered at each workstation (since in order to trigger new device activities, no external input is required to confirm the execution of the device activity). In this context, go-to templates are not applicable.
Device Route Activity—Restriction $DVRA_{RESTRICTION}$ DVRA ID: — / **Restriction**: VP2 < 100	Devices that have restrictions to execute specific device route activities at a workstation—due to equipment characteristics or the layout of the manufacturing system. For instance, if pick and load activities have a restriction defined, namely, VP2 < 100, that means only volumes whose volume parameter value of VP2 is lower than 100 can be loaded. VP2 can represent the weight or the space required for a specific volume. This property can apply to any AGV, forklift or truck—mostly due to restrictions related to the dimensions or weight of the object.

Regarding the representation of a TMHS, Figure 14 shows an example with the following features:

1. TMHS has three devices (DV1 to DV3) and eight workstations (WK1 to WK8);
2. Each device is defined by a set of device route links (identified with the same color) and activities. The properties that characterize each device route link or activity can be applied and modified as presented in Table 9. The sequence of device route links and activities of device DV1 is the same as presented in Figure 13;
3. There are two workstations where objects can be transferred—WK6 and WK8. WK6 is a transfer point, and WK8 is a meeting point.

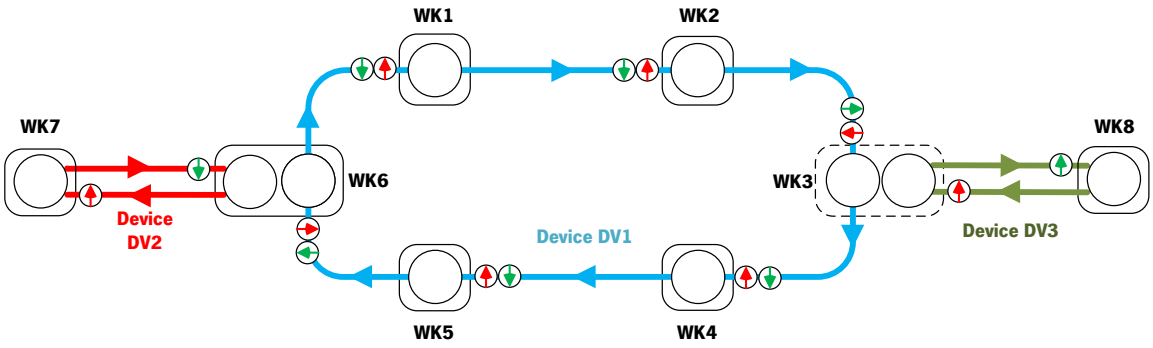


Figure 14. Representation of a TMHS.

Like the device route discussed above, the representation of the TMHS shown in Figure 14 can be applied to several examples. Some particularities and examples are pointed out here—in particular, to show the application of transfer points and meeting points:

- Workstation WK6 is a transfer point where devices DV1 and DV2 can load and unload volumes in order to be transferred between these devices. This representation can be applied to any set of devices that can unload objects temporally at a transfer point to be transported by another device—for instance, to represent the transfer of objects between different tugger trains, between a train and trucks, or AGVs and automated storage and retrieval equipment;
- Workstation WK3 is a meeting point where devices DV1 and DV3 should synchronize their activities in order to transfer objects between them. This representation can be applied to any set of devices that must be placed at the same meeting point in order to load an object from one device to another—for instance, to represent the transfer of objects between cranes and trucks (to handle a heavy load, such as a coil) or between an automated robot arm and any device where the object should be placed.

4.2. Representation of Case Studies from the Literature

The main purpose of this section is to represent some case studies selected from the literature with the proposed language and compare the information provided and represented in each option.

The case studies considered are as follows:

- Automated material delivery system: a study conducted by Rahman & Nielsen [28];
- Freight interchange site: a study conducted by Gattuso & Cassone [10];
- Vehicle with different distribution strategies: a study conducted by Hoff, Gribkovskaia, Laporte & Løkketangen [25].

4.2.1. Automated Material Delivery System

The first example refers to a representation of an "automated material delivery system" composed of automated transport vehicles (ATVs) on a shop floor. Rahman & Nielsen define an ATV as automatic equipment "used to transport materials to each machine on the production floor and to load containers into container terminals in seaports". Some listed examples are AGV, mobile robots (or self-driving vehicles), tow trains or unmanned aerial vehicles [28].

The aim of Rahman & Nielsen was to present a new methodology to schedule activities for automated transport and handling equipment. In their paper, they describe two systems—a shop floor and an automated container terminal [28].

Figure 15 shows a typical layout of an automated shop floor. The shop floor is made up of machines or workstations where materials should be delivered to execute manufacturing jobs and two warehouses where materials must be prepared to be delivered to machines. The figure also represents pickup points at each warehouse, delivery points at each machine and two types of devices—ATVs and a conveyor belt (for semi-finished products) [28].

Transportation and handling activities are triggered by manufacturing jobs. "When production starts, some machines need some materials to continue production." An ATV goes to a warehouse in order to pick up the required materials. Then, the ATV "travels to the designated machine where the materials need to be delivered and delivers the material to the delivery point" [28].

Figure 16 represents the automated material delivery system presented.

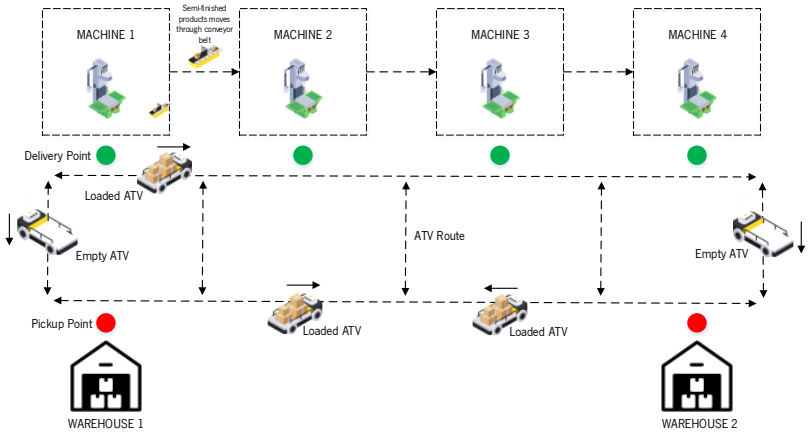
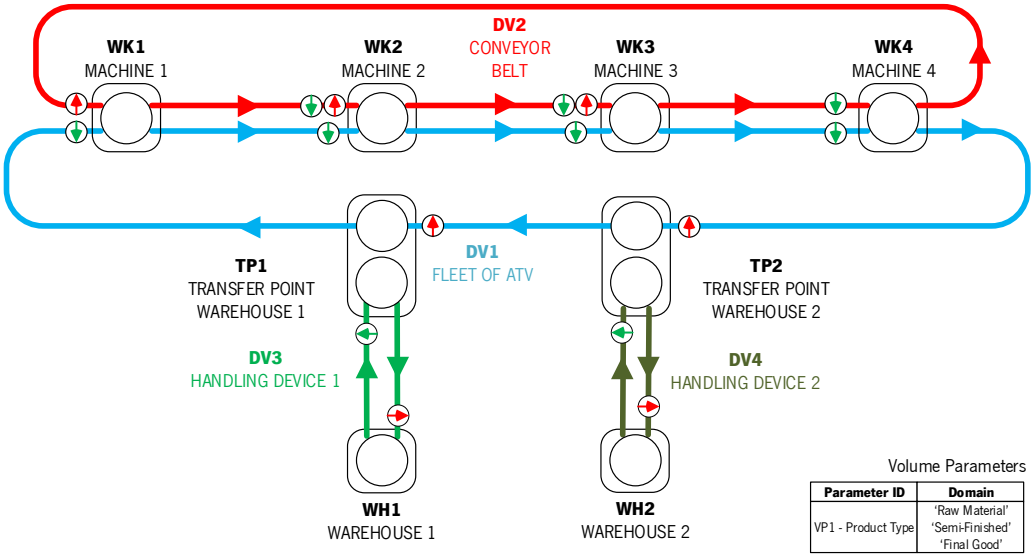


Figure 15. Automated material delivery system with ATV on a shop floor (adapted from [28]).



Volume Parameters

Parameter ID	Domain
VP1 - Product Type	'Raw Material' 'Semi-Finished' 'Final Good'

Additional Information of each device route link

Device	DVRL ID	Origin WK	Destination WK	Time	Execution Entity	Go-To Template	Restriction
DV1	1	TP1	WK1	5	Automatic		VP1='Raw-Material'
DV1	2	WK1	WK2	5	Automatic		VP1='Raw-Material'
DV1	3	WK2	WK3	5	Automatic		VP1='Raw-Material'
DV1	4	WK3	WK4	5	Automatic		VP1='Raw-Material'
DV1	5	WK4	TP2	5	Automatic		VP1='Raw-Material'
DV1	6	TP2	TP1	5	Automatic		VP1='Raw-Material'
DV2	1	WK1	WK2	5	Automatic		VP1='Semi-Finished'
DV2	2	WK2	WK3	5	Automatic		VP1='Semi-Finished'
DV2	3	WK3	WK4	5	Automatic		VP1='Semi-Finished'
DV2	4	WK4	WK1	5	Automatic		VP1='Semi-Finished'
DV3	1	WH1	TP1	5	Device		
DV3	2	TP1	WH1	5	Device		
DV4	1	WH2	TP2	5	Device		
DV4	2	TP2	WH2	5	Device		

Additional Information of each device route activity

Device	DVRL	DVRA ID	Action Type	Synch. Mode	Execution Entity	Input Required	Handling Template	Restriction
DV1	1	1	Unload and Store	On Arrival	Device	No	Template A	
DV1	2	1	Unload and Store	On Arrival	Device	No	Template A	
DV1	3	1	Unload and Store	On Arrival	Device	No	Template A	
DV1	4	1	Unload and Store	On Arrival	Device	No	Template A	
DV1	5	1	Pick and Load	On Arrival	Device	No	Template B	
DV1	6	1	Pick and Load	On Arrival	Device	No	Template B	
DV2	1	1	Unload and Store	On Arrival	Device	No	Template C	
DV2	1	2	Pick and Load	On Arrival	Device	No	Template C	
DV2	2	1	Unload and Store	On Arrival	Device	No	Template C	
DV2	2	2	Pick and Load	On Arrival	Device	No	Template C	
DV2	3	1	Unload and Store	On Arrival	Device	No	Template C	
DV2	4	1	Pick and Load	On Arrival	Device	No	Template C	
DV3	1	1	Unload and Store	On Arrival	Device	Yes	Template D	
DV3	2	1	Pick and Load	On Arrival	Device	Yes	Template D	
DV4	1	1	Unload and Store	On Arrival	Device	Yes	Template E	
DV4	2	1	Pick and Load	On Arrival	Device	Yes	Template E	

Figure 16. Representation of an automated material delivery system.

The representation of Figure 16 reflects the interpretation of the authors of data available in the paper conducted by Rahman & Nielsen—which shows the difficulties of representing TMHS with existing approaches and the ability of the language to represent each element of TMHS. Some examples of information interpreted by the authors are:

- The fleet of ATVs is represented only with one device—but each ATV of the fleet could also be represented as a single device. The figure considers a fleet since no information has been identified about each ATV;
- Apart from the representation of the fleet of ATVs and the conveyor belt (also represented in Figure 15), Figure 16 considers two transfer points and handling equipment to warehouses 1 and 2. The authors stated that "to transfer those materials, an ATV goes to the transfer point of a warehouse and picks up material" [28]. To represent this behavior, the authors assumed two pieces of handling equipment (Handling Devices 1 and 2) to pick the right volume from the warehouse and place it at the respective transfer point—in order to be available when an ATV arrives at the transfer point.
- Other options could have been considered (but are not represented in Figure 16), for instance, ATVs may transport storage locations to machines or ATVs may have handling equipment to execute themselves picking, loading or unloading tasks;
- Figure 16 considers that an ATV only transports raw materials from warehouse transfer points to machines since this information can be deducted from Figure 15 and the respective paper. Nevertheless, it is not clearly defined in Figure 15 and the paper, for instance, if other product types (such as semi-finished or a tool) can also be transported from the warehouse to machines;
- Figure 15 and the paper do not present information about the return of materials—since it is not defined if an ATV can also load volumes at machines and return them to warehouses. Figure 16 considers that an ATV can only load at warehouse transfer points and unload at machines;
- The authors also stated information about the management of weight capacity. "Each task requires a certain amount (weight) of materials to be delivered. Therefore, an ATV may pick up and deliver materials for multiple tasks, if it remains within its maximum carrying capacity in a tour" [28]. This information is not represented in Figure 16 because more information is required, but it can also be represented based on restrictions associated with a device route link or activity.

Moreover, many assumptions have been made since no information was found in the paper about:

- The device route activities that can be executed at each workstation by each device. Figure 16 considers that an ATV can only load volumes at transfer points and unload them at machines, and that handling equipment should load volumes from the warehouse and unload them at the transfer point. Other scenarios could also be considered—such as loading and unloading at all workstations;
- The interaction and synchronization rules between devices—for instance, the interaction between warehouse handling equipment and ATVs are represented as transfer points—could also be meeting points. No information about it was found in the paper;
- Who must execute device route activities—the authors assumed that activities are always executed by devices; nevertheless, no evidence was found to argue that;
- The type of synchronization between devices and workstations—the representation proposed considers on-arrival synchronization mode;
- If several device route activities can be triggered for each device at the same time—the representation of Figure 16 proposes that all ATV fleet and conveyor device route links should be executed automatically and that device route activities do not require an

external input to trigger new activities (because this behavior is common on these types of equipment);

- If any equipment requires instructions—the representation proposed does not consider go-to templates, but considers handling templates;
- The time to execute each device route link—5 is considered for all device route links.

4.2.2. Freight Interchange Site—From Ship to Train

Inbound and outbound logistics may involve transferring objects (mostly containers) from one device to another, such as from a ship to a train or from a train to a truck, in order to get to the final destination.

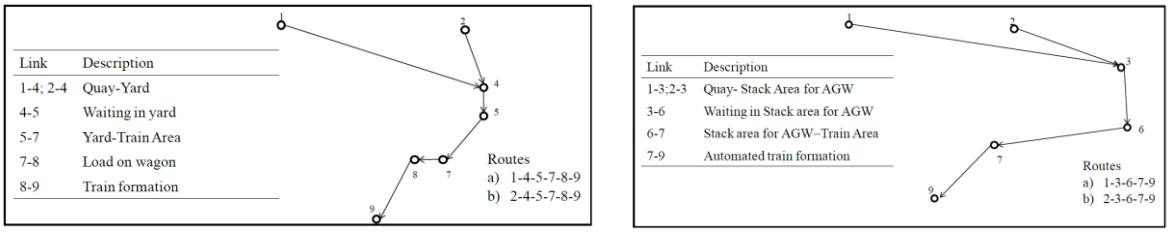
Gattuso & Cassone carried out a study to analyze and quantify the benefits of Automated Guided Wagon (AGW)—to handle and transport containers in ports. They conducted their study to analyze specifications related to "spatial, organizational and relational structure" and represent existing "links and paths" based on a freight interchange site [10].

One of the case studies analyzed by Gattuso & Cassone is the Gioia Tauro port. Table 10 presents all alternatives for freight transfers from the quay to the train area considering several types of transport and handling equipment—namely, AGV, AGW, gantry cranes (GCs), straddle carriers (SCs), reach stackers (RS), transtainers (TTs) and multitrailer trains (MTs) [10].

Table 10. Alternative paths (1, 2, 3 and 4) [10].

	1	2	3	4
Ship unloading	GC	GC	GC	GC
Transfer to yard	SC	SC	AGV	AGW
Transfer to intermodal terminal	SC	MT+RS	AGV+RS	AGW
Train loading	RS	TT	TT	-

Figure 17a presents the graph of alternatives 1, 2 and 3—considering conventional equipment and AGV. Figure 17b presents the graph of alternative 4—considering AGW.



(a) **(b)**

Figure 17. Graph for (**a**) Alternatives 1, 2 and 3 (conventional equipment and AGV); (**b**) Alternative 4 (AGW) [10].

In order to compare both approaches to represent TMHS, Figure 18 represents the freight interchange site of Gioia Tauro port. It should be noted that, again, the representation reflects the authors' interpretation of the data presented in the paper. Many assumptions have been made due to the difficulty of the authors to represent each element of the TMHS—since it was difficult to accomplish the following:

- Identify all possible devices, device route links and activities of the TMHS;
- Identify all action types that can be executed at each workstation by each device;
- Define interaction and synchronization rules between devices—for instance, it can be read in the paper that "at the yard blocks there are racks where the automated

vehicle can deposit the container without waiting for the crane to pick it up" [10]. Nevertheless, this information is not expressed in Figure 17, and no more information is provided regarding the remaining workstations where objects can be transferred between devices (such as in the intermodal terminal);

- Define synchronization rules with workstations—for instance, it is not possible to identify who must execute loading and unloading activities at each workstation, when the device can unload volumes at each workstation, if an input is required to trigger new activities and so on;
- Define restrictions for device route links and activities—for instance, some equipment may load just a container at a time, or some devices may have associated restrictions to load objects related to size, weight or volume.

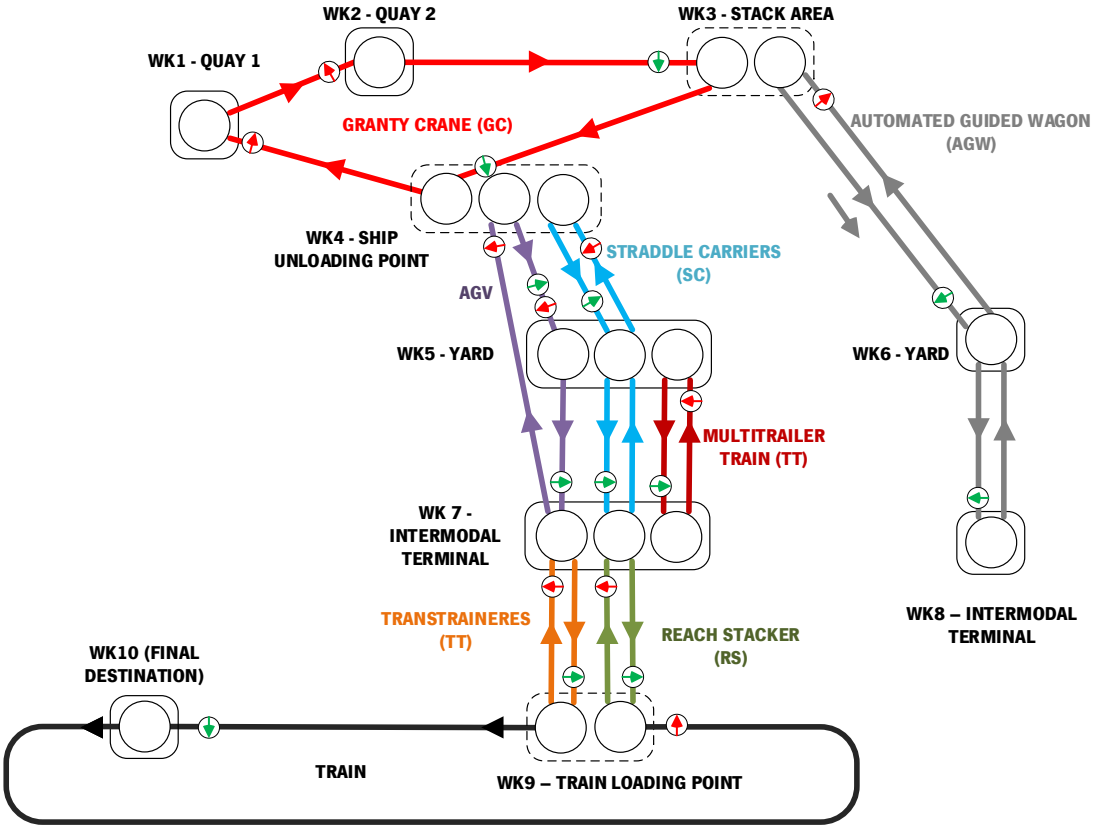


Figure 18. Representation of a freight interchange site.

4.2.3. Vehicles with Different Distribution Strategies

Hoff, Gribkovskaia, Laporte & Løkketangen conducted a study where some routes of devices were identified as "vehicle routing problem with pickups and deliveries"—among others, they present Hamiltonian and Double-path routes (Figure 19). Hamiltonian and Double-path routes can be represented through the same set of workstations to be visited, but the activities that can be executed at each workstation vary [25]:

- Hamiltonian distribution strategy considers that, at each workstation, the device can deliver and collect volumes (during the same visit) (Figure 19a);
- Double-path strategy considers that first all deliveries are executed and just then, in the reverse order, volumes are collected at each workstation (Figure 19b).

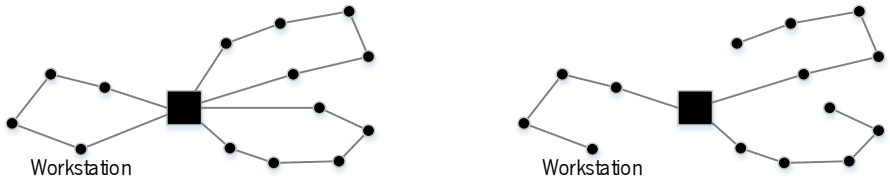


Figure 19. Scheme of routes: (**a**) Hamiltonian and (**b**) Double-path (adapted from [25]).

Figure 20a,b represent, respectively, Hamiltonian and Double-path strategies with the proposed language, showing the language's ability to represent different shapes of device routes and distribution strategies.

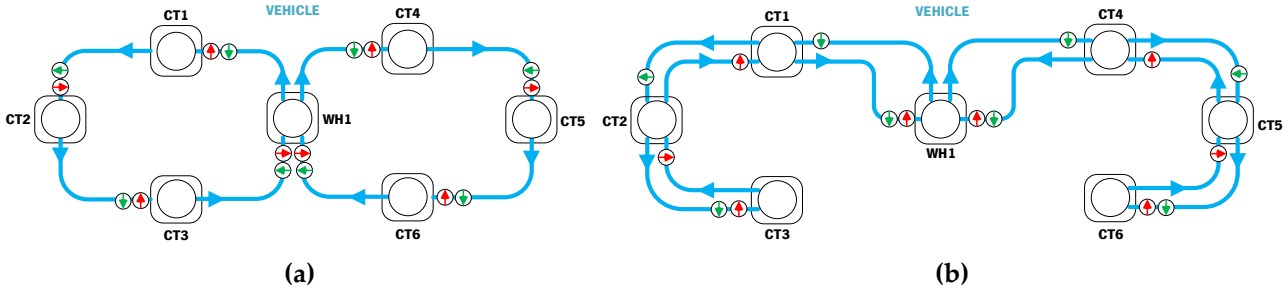


Figure 20. Representation of two different vehicle routes: (**a**) Hamiltonian distribution strategy; (**b**) Double-path distribution strategy.

It should be noted that the authors could only distinguish the behavior of each distribution strategy in the study because this information is described in the paper [25]. Considering only the data in Figure 19, it is not possible to conclude if volumes can or cannot be delivered and collected at each workstation. Considering Figure 20, this information is clear since loading and unloading device route activities are identified at each device route link.

5. Discussion

The generic representation language proposed consists of only three element types (objects, workstations and devices) and a small set of simple and easy-to-use properties to be defined by users of each organization to characterize each element of a TMHS. Each property is not related to any device in particular but can be used to represent the behavior of different devices. In total, around 26 properties have been identified to characterize different elements of TMHS and 16 to define their behaviors. Moreover, if necessary, new properties can be added to the representation language in future work to represent any other characteristic of an element of a TMHS.

A graphic representation for each element is also proposed to facilitate the representation of TMHS since it creates a clear, concise and objective language that standardizes the communication between users, allowing the schematization of the network of devices, the representation of information needed to characterize each element of a TMHS and the validation of the behaviors defined by users for each element. The graphic representation consists of four different symbols and a set of optional tables that define the remaining properties that must be identified by users to reflect in the best way the real behavior of each element of the TMHS, which shows the simplicity, ease and speed with which users can represent the TMHS of the organization. In addition to reducing the effort required to represent each TMHS, this approach also reduces the learning time and effort to apply this language, since the effect of each property on the manufacturing system is clearly defined. The representation language proposed also helps users of each organization to

define the rules and behaviors of the TMHS of each organization, as the set of properties identifies important points to take into consideration instead of starting from scratch, and comparisons can be made with other examples. All these points are extremely important to ensure the proper representation of the elements, rules and restrictions that characterize each TMHS since, after its implementation in an IT support system, they should influence how transport and handling activities are controlled.

The representation of several examples throughout this paper shows the ability and flexibility of the language to represent different industrial devices and TMHS, and that similar representations can be applied to different types of equipment. The strategy to test and validate the representation language considered two types of scenarios: the first section considers some examples of device representations to identify which industrial transport/handling equipment could fit at each representation, and the other sections consider the representation of specific devices of a TMHS, namely, the representation of the case studies from the literature and from the industry.

The representation of three examples (case studies) from the literature with the proposed language, and the comparison of the information provided and represented in each option, also revealed interesting results. The authors had to make many assumptions to represent these examples, for instance, by deducing data from the text in the respective articles. This shows not only the difficulty of representing transport and handling device behaviors based on existing models and techniques, but also the ability of the developed language to represent the characteristics and behaviors of real examples of TMHS.

It is important to note that all selected examples represent the characteristics of a specific TMHS, but they did not aim to develop a generic and comprehensive language to represent TMHS. Therefore, the study objectives and requirements are not the same. No representation language has been found capable of representing different TMHS effectively. Authors applied traditional data modeling techniques such as flow charts or diagrams, probably because they did not find a better and standard tool to represent each TMHS.

The concepts of template and template parameters are also considered to allow users to define the form to trigger instructions to industrial equipment, systems or even user interfaces, which is aligned with the requirements of Industry 4.0 and Smart Manufacturing Systems. The template parameters proposed for each template (go-to or handling template) refer to the main information that was been recognized as necessary to be sent to an equipment. However, other template parameters can be added in the future considering any information available. For instance, some equipment requires information of the part number to be handled (when the equipment management system can identify the volume from where items should be picked), and this information can be represented as a volume parameter and added in the future as a template parameter. The properties that identify the physical location of workstations, storage locations, storage location positions or device positions can also be added in the future as template parameters.

Despite obtaining positive results, some limitations can be pointed to in the ability of the proposed language to represent some particularities of TMHS, namely:

- Organizations can only define one template for each device route link or activity. However, there are cases where multiple instructions need to be sent based on the same device route activity at the same time. For example, it can be necessary to trigger automatic tasks for a powered roller conveyor of an AGV and an external platform of the workstation.
- Each object handled in a device route activity can only have one instruction defined. Some equipment, like an AGV with a lift, may only need one instruction when arriving or departing from a workstation. In these cases, each object handled will trigger one instruction when only one is needed.

- Devices do not follow a fixed time schedule. Some devices, such as airlines, motor carriers, cruise ships, bus lines, railroads and freight services, have schedules that dictate their departure and arrival times at different workstations along their route. These schedules determine the time each device can reach a workstation.

The first and second limitations can be overcome because organizations can identify other ways of dealing with these behaviors without adding more properties that need to be always defined by users, such as triggering multiple tasks based on one instruction or deleting duplicated instructions. The last behavior has not been considered because it would increase significantly the complexity of the representation language and that does not represent the behavior of most of the devices.

Finally, it is also important to note that the properties that represent restrictions (for storage location position, device position, device route link and device route activity) increase the flexibility and comprehensiveness of the language, as they allow users of each organization to define and represent their specific rules. However, they also increase the effort to implement these properties in an IT support system. This feature of the representation language will require the development of a mathematical expression parser where volume and storage location parameters can be defined by users as operands, which implies, among other things, the validation of the domain of each parameter and the implementation of different functions (for instance, the function to validate the total weight that can be loaded in a device should consider the weight of all objects already loaded in the device and the weight of the object that needs to be loaded).

6. Conclusions

One of the main objectives of Industry 4.0 and Smart Factories is the control of all manufacturing system entities, not only regarding medium-term planning, but also the synchronization and scheduling of the activities of all resources, whether productive resources (such as execution workstations), transport equipment, storage systems, etc.

This paper presents a generic language to represent and control TMHS in manufacturing systems. The language can be used to represent TMHS with multiple devices that transport and handle objects in discrete manufacturing systems. A device can be a single piece of equipment but also a group of equipment (such as a set of AGVs with the respective fleet management platform).

The main contribution of the proposed language, when compared with traditional approaches in the literature, is that it can be effectively applied in different industry types and with different equipment to represent the TMHS of each organization. The generic language allows the representation of a TMHS based on three element types and a set of simple properties that can be easily and quickly defined by users of each organization to achieve the following:

- Characterize objects that can be transported by a device;
- Characterize workstations, transfer points and meeting points of the manufacturing system, and all physical areas at each workstation where a volume can be placed;
- Characterize the behavior of each device in a manufacturing system—including the physical areas where objects can be loaded and device route links/activities that the device can execute. Each device is represented by a route that includes properties to define what links can be executed, what device route activities can be executed (action type property), when, by whom (execution entity), the synchronization rules between devices (workstation type), synchronization rules with workstations of the manufacturing system (synchronization mode/input required) and restrictions that represent different rules to limit which activities can be assigned to a device.

- The interface with industrial transport and handling can also be defined based on templates capable of defining the form to trigger the information needed to define a specific instruction to industrial equipment, according to the function and data structures that need to be sent to the equipment.

To the best of the authors' knowledge, no other language is known to be capable of representing several devices of manufacturing systems, and the initial purpose of the representation language development was to fill this gap. The outcomes of the test and validation phase show the comprehensiveness and flexibility of the language to represent the specific needs of several TMHS, including the representation of different links, workstations, activities, objects, transfer points, storage places, rules for handling volumes, etc. The main features of the representation language that can be pointed out are as follows:

- Generic—the language eliminates the need to define elements and properties to represent all distinct devices and activities. The language allows the representation of several characteristics of a TMHS and its devices, which does not mean having a model to represent each specific TMHS or industrial equipment. All properties of the language are not related to any specific device but can be used to represent the behavior of different devices;
- Simple, easy and quick to use;
- Provides more and better information—which is even more important in the actual context of Industry 4.0 and Smart Factories, as the quantity of information and functionalities increased to satisfy the Smart Manufacturing System requirements;
- Flexible and adaptable—the language applies to various organizations and industries. It supports the representation of new transport and handling equipment in manufacturing systems, whether manual, semi-automatic or fully automatic. It can also be used to represent any natural adjustments or changes in the TMHS.

Finally, in order to increase the outcomes of the work carried out, some future work should be considered. As referred to in the introduction, this work is part of an ongoing research project that aims to develop a generic model for representing and controlling TMHS in Smart Manufacturing Systems. The model is composed of static information that represents each element of the TMHS (presented in this paper) and a set of processes to control transport and handling activities of each device and synchronize them with the remaining activities of manufacturing systems. These processes must use the information provided by the representation language, but also imply the definition of a set of structures for storing dynamic information that varies according to the system status (in contrast to properties presented in this paper, these properties are not defined by users of organizations to represent a particular element, and they should be managed by the processes of the model). The processes should be able to control TMHS characterized by several devices and deal with multiple requests to transport materials simultaneously. The representation language presented in this paper should provide all the needed information so that processes can be implemented to control transport and handling activities in several organizations. Moreover, the representation language is not a closed development, and new properties can be added to represent any characteristic of an element of a TMHS or to represent information required for the processes referred to previously.

Author Contributions: Conceptualization, M.G. and P.M.; methodology, M.G., P.M. and G.P.; validation, P.M. and R.S.; formal analysis, M.G., P.M., R.S. and G.P.; investigation, M.G., P.M. and G.P.; resources, M.G., P.M. and G.P.; data curation, M.G.; writing—original draft preparation, M.G.; writing—review and editing, P.M. and R.S.; visualization, P.M., G.P. and R.S.; supervision, P.M. and G.P.; project administration, P.M. and G.P.; funding acquisition, P.M. and G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Portuguese Foundation for Science and Technology (FCT), grant number PD/BD/140854/2018.

Data Availability Statement: Data are contained within the article.

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

References

1. Kagermann, H.; Wahlster, W.; Helbig, J. *Recommendations for Implementing the Strategic Initiative Industrie 4.0: Final Report of the Industrie Working Group*; National Academy of Science and Engineering: Washington, DC, USA, 2013.
2. Lucke, D.; Constantinescu, C.; Westkämper, E. Smart Factory—A Step towards the Next Generation of Manufacturing. In *Manufacturing Systems and Technologies for the New Frontier*; Springer: London, UK, 2008; pp. 115–118. [CrossRef]
3. Goncalves, M.; Martins, P.; Pereira, G. Challenges to Represent and Manage Transport and Material Handling Systems in Manufacturing Systems. In Proceedings of the 2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 18–21 December 2023; pp. 1507–1511. [CrossRef]
4. Alves, J.; Lima, T.M.; Gaspar, P.D. Is Industry 5.0 a Human-Centred Approach? A Systematic Review. *Processes* **2023**, *11*, 193. [CrossRef]
5. Lin, I.C.; Tseng, P.C.; Chen, P.H.; Chiou, S.J. Enhancing Data Preservation and Security in Industrial Control Systems through Integrated IOTA Implementation. *Processes* **2024**, *12*, 921. [CrossRef]
6. Yan, J.; Zhang, M.; Fu, Z.; Yan, J.; Zhang, M.; Fu, Z. An Intralogistics-Oriented Cyber-Physical System for Workshop in the Context of Industry 4.0. *Procedia Manuf.* **2019**, *35*, 1178–1183. [CrossRef]
7. Li, M.; Guo, D.; Li, M.; Qu, T.; Huang, G.Q. Operation Twins: Production-Intralogistics Synchronisation in Industry 4.0. *Int. J. Prod. Res.* **2022**, *61*, 5193–5211. [CrossRef]
8. Knoll, D.; Reinhart, G.; Prüglmeier, M. Enabling Value Stream Mapping for Internal Logistics Using Multidimensional Process Mining. *Expert Syst. Appl.* **2019**, *124*, 130–142. [CrossRef]
9. Efthymiou, O.K.; Ponis, S.T. Current Status of Industry 4.0 in Material Handling Automation and In House Logistics. *Int. J. Ind. Manuf. Eng.* **2019**, *13*, 1370–1374.
10. Gattuso, D.; Cassone, G.C. AGW for Efficient Freight Transport in Container Yard: Models and Costs. *Transp. Res. Procedia* **2018**, *31*, 108–120. [CrossRef]
11. Alnahhal, M.; Noche, B. Efficient Material Flow in Mixed Model Assembly Lines. *Springerplus* **2013**, *2*, 415. [CrossRef] [PubMed]
12. Campos, T.; Carvalho, M.; Oliveira, J.A.; Silva, P.V.; Machado, T. Using Discrete Simulation to Support Internal Logistics Process Design. In Proceedings of the 47th International Conference on Computers & Industrial Engineering 2017 (CIE47), Lisbon, Portugal, 11–13 October 2017; pp. 11–13.
13. Kluska, K.; Pawlewski, P. The Use of Simulation in the Design of Milk-Run Intralogistics Systems. *IFAC-PapersOnLine* **2018**, *51*, 1428–1433. [CrossRef]
14. Sulírová, I.; Závodská, Ľ.; Rakyta, M.; Pelantová, V. State-of-the-Art Approaches to Material Transportation, Handling and Warehousing. *Procedia Eng.* **2017**, *192*, 857–862. [CrossRef]
15. Wu, C.; Xiao, Y.; Zhu, X. Research on Optimization Algorithm of AGV Scheduling for Intelligent Manufacturing Company: Taking the Machining Shop as an Example. *Processes* **2023**, *11*, 2606. [CrossRef]
16. Parvini, M. 9—Packaging and Material Handling. In *Logistics Operations and Management*; Farahani, R.Z., Rezapour, S., Kardar, L., Eds.; Elsevier: London, UK, 2011; pp. 155–180, ISBN 978-0-12-385202-1.
17. Martins, P.; Gonçalves, M.; Pereira, G. The Importance of Integrating Transport and Handling Activities in Manufacturing Planning and Control Systems. In Proceedings of the ICPR025 2021: XV. International Conference on Production Research, New York, NY, USA, 18–21 July 2021; pp. 57–67.
18. Dang, Q.V.; Nguyen, C.T.; Rudová, H. Scheduling of Mobile Robots for Transportation and Manufacturing Tasks. *J. Heuristics* **2019**, *25*, 175–213. [CrossRef]
19. Gonçalves, M.; Martins, P.; Pereira, G. Element Types and Properties to Represent Transport and Material Handling Systems in Manufacturing Systems. In *Proceedings of the CIE51 Proceedings*; Elsevier: Sydney, Australia, 2024.
20. Oosthuizen, M.J.H. Chapter 9—Action Research. In *Research Methods for Students, Academics and Professionals*, 2nd ed.; Williamson, K., Bow, A., Burstein, F., Darke, P., Harvey, R., Johanson, G., McKemmish, S., Oosthuizen, M., Saule, S., et al., Eds.; Topics in Australasian Library and Information Studies; Chandos Publishing: Oxford, UK, 2002; pp. 159–175, ISBN 978-1-876938-42-0.
21. Plattform Industrie 4.0—RAMI4.0—A Reference Framework for Digitalisation. 2019. Available online: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.html (accessed on 18 December 2024).
22. Sadjady, H. 2—Physical Flows. In *Logistics Operations and Management: Concepts and Models*; Farahani, R.Z., Rezapour, S., Kardar, L., Eds.; Elsevier: London, UK, 2011; pp. 11–40, ISBN 978-0-12-385202-1.

23. Carlos, J.; Alice, A.; Rui, M.; J, M.P. Semantic Mediation: A Literature Review on Semantic Interoperability through Ontologies. In Proceedings of the International Conference on Industry Sciences and Computer Science Innovation Semantic, Amman, Jordan, 29–31 October 2024; pp. 1–10.
24. Directorate-General for Digital Services (European Commission). *European Interoperability Framework (EIF)—Towards Interoperability for European Public Services*; Publications Office of EU: Luxembourg, 2011.
25. Hoff, A.; Gribkovskaia, I.; Laporte, G.; Løkketangen, A. Lasso Solution Strategies for the Vehicle Routing Problem with Pickups and Deliveries. *Eur. J. Oper. Res.* **2009**, *192*, 755–766. [CrossRef]
26. Zhang, Y.; Zhang, G.; Du, W.; Wang, J.; Ali, E.; Sun, S. An Optimization Method for Shopfloor Material Handling Based on Real-Time and Multi-Source Manufacturing Data. *Int. J. Prod. Econ.* **2015**, *165*, 282–292. [CrossRef]
27. Tran, N.K.; Haasis, H.D.; Buer, T. Container Shipping Route Design Incorporating the Costs of Shipping, Inland/Feeder Transport, Inventory and CO2 Emission Oa. *Marit. Econ. Logist.* **2017**, *19*, 667–694. [CrossRef]
28. Rahman, H.F.; Nielsen, I. Scheduling Automated Transport Vehicles for Material Distribution Systems. *Appl. Soft Comput. J.* **2019**, *82*, 105552. [CrossRef]

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.