

A Rule-Based Approach Founded on Description Logics for Industry 4.0 Smart Factories

Georgios Kourtis, Evangelia Kavakli, and Rizos Sakellariou, *Member, IEEE*

Abstract—This paper develops a formal framework, founded on description logics, to assist decision making in relation to the manufacturing operation and control in modern enterprises that stand to benefit from the transition to Industry 4.0. The objective is to provide sophisticated support to individuals making decisions in the area of production operations management and in particular production scheduling and material requirements planning. Using this framework, the paper demonstrates an approach to encode the domain knowledge of human experts managing the production as sets of formal rules. These rules can be implemented in an intelligent system that can assist and empower human experts, reducing difficulty when making decisions in complex manufacturing environments.

Index Terms—Industry 4.0, description logics, industrial production modelling.

I. INTRODUCTION

Modern industrial production is undergoing radical changes, due to the advent of new technologies such as cloud computing [40], the Internet of Things [12], and cyber-physical systems [27]. The emerging new ways of production are usually classified under the term *Industry 4.0* [20]. One of the motivating challenges of Industry 4.0 is to provide faster time to market while responding to a rise in consumer demand. This requires an optimal utilization and orchestration of industrial resources as well as quick response to disruptions in production.

Traditionally, decision making in relation to production and resource management relies on human experts who are supported by specialized software, such as business processes management [42], database and data analytics [30], or enterprise resource management software [14], [38]. Despite the abundance of such software, the complexity of modern industrial tasks combined with an exponential growth of available information related to production (e.g. due to advances in computer-integrated manufacturing and cyber-physical systems) makes it exceedingly difficult for humans

Georgios Kourtis is with the Department of Computer Science, University of Liverpool, Liverpool L69 3BX, United Kingdom (e-mail: G.Kourtis@liverpool.ac.uk).

Evangelia Kavakli is with the Department of Cultural Technology and Communication, University of the Aegean, Mtilini 81100, Greece (e-mail: kavakli@ct.aegean.gr) and with the School of Computer Science, The University of Manchester, Manchester M13 9PL, United Kingdom (e-mail: evangelia.kavakli-2@manchester.ac.uk).

Rizos Sakellariou is with the School of Computer Science, The University of Manchester, Manchester M13 9PL, United Kingdom (e-mail: rizos@manchester.ac.uk).

Manuscript received November 20, 2017; revised June 20, 2018; accepted April 10, 2019.

The work in this paper was partially supported by the H2020 project DISRUPT (project no 723541). Georgios Kourtis would also like to acknowledge support from the FAIR-SPACE EPSRC project (EP/R026092/1).

to make optimal decisions. Nevertheless, the knowledge of experts remains an indispensable part of production planning. It is sensible, then, to consider whether the domain knowledge of human experts responsible for organizing the production can be systematized and incorporated into intelligent systems, so as to enable sophisticated decision making in complex settings.

We argue that any attempt for a systematic representation of the knowledge in the above domain would benefit from recent advances in knowledge representation. For, any intelligent system that aims to encapsulate human expert knowledge has to be able to perform *reasoning*. Reasoning, in the context of intelligent systems, is the process in which a system automatically makes logical deductions based on rules and facts in a given universe of discourse. In general, reasoning problems are known to be among the hardest when it comes to their computational complexity [7]. Thus, apart from providing an effective way to represent and store knowledge, an ad hoc intelligent system for the management of production also has to implement an optimized reasoner, which is an error-prone and difficult task [4], [5]. It is our view that to facilitate the transition to the Industry 4.0 vision, manufacturing enterprises can benefit from a generic, standardized formalism for the description of their organizational structure with regard to production. Most importantly, such a formalism could lay the foundations for the implementation of an intelligent production management system, i.e., a system capable of reasoning based on facts and sets of predefined rules, an application area that has not yet received much attention. Unfortunately, at the moment, there is no widespread formalism of this kind.

Existing approaches (e.g. [8], [19], [29]) mainly focus on describing the entities involved in production and their interactions; that is, their main concern is *modeling* rather than how modeling can enable reasoning. Although modeling is helpful for humans to structure their understanding of a given domain, it is not sufficient on its own to assist in decision making. In this respect, the main focus in this paper is to model knowledge in such a way that it enables reasoning. Our intention is to lay the foundations for an automated system which, based on predefined facts and rules set by domain experts, can make logical deductions and, by those deductions, assist decision making in a modern manufacturing environment, also catering for situations of disruption in the production or in the supply chain.

To this end, our contribution is twofold. First, we demonstrate how to build a formal framework based on *description logics* [4]. To keep the size of the problem manageable, we limit our scope to two aspects of production management, namely production scheduling and material requirements

planning. Special attention is placed on the modeling of the temporal progression of production, while the use of description logics increases the practical effectiveness of the proposed framework as it can be relatively easily implemented. Second, we prescribe how to build, using the above formalism, a knowledge base (comprising various facts and rules) based on which a reasoner can derive, for a given production environment, the actions that need to be performed at a given point in time or in response to an event. Using our framework, a manufacturing enterprise whose production is managed by a team of experts can encode the domain knowledge of these experts as sets of rules and, based on these rules, delegate many important decisions to an intelligent system. Such decisions may range from simple tasks (such as arranging to order a material when it runs out) to complex responses to events (such as dynamically distributing orders for a given material to new suppliers when a given supplier is unable to fulfill a request). The proposed approach consolidates and generalizes the requirements of two end-users from the automotive and the white goods sector in the context of the EU-funded H2020 project DISRUPT¹. The project aims to develop a comprehensive data-driven solution for automated vertical and horizontal integration, facilitating the transition into *smart manufacturing* [25]. It has a specific focus on decision making in case of disruptions in production.

Description logics [4] were chosen as the formal underpinning of our approach because of their wide use in knowledge representation. These languages are expressive enough to model complex situations while retaining good practical performance (and reasonable worst-case performance compared to other expressive formal languages such as fragments of first-order logic). They also support various query languages and so-called justifications [21], i.e., human-readable explanations of deduced facts. Thus, monitoring and understanding the actions of an intelligent system based on description logics is relatively straightforward. In addition, there is an ongoing line of research to adapt description logics for reasoning with uncertainty (probabilistic description logics [35]), which may turn out to be useful in the industrial setting, as modern production environments involve various degrees of uncertainty, often making these environments hard to model with deterministic formal languages.

The remainder of the paper is structured as follows. In Section II we discuss related, knowledge-driven approaches to modeling various aspects of industrial production using formal languages. Section III describes the abstract model of production on which our approach is based and presents a general architecture for the type of intelligent system that we propose. Section IV demonstrates how to build a knowledge base using the above formalism. Section V discusses how to extend the above knowledge base with rules to form a complete intelligent system and give some examples of rules for three specific use cases. Finally, Section VI concludes the paper and discusses possible extensions of this research.

II. RELATED WORK

It has long been understood that the high complexity of modern industrial tasks makes it exceedingly difficult for humans to organize and supervise the production in modern enterprise environments. As a result, there have been various attempts to formalize different aspects of production, aiming to assist humans in their decisions and also as a first step towards automating human tasks. Yet, manufacturing decisions still rely heavily on the knowledge of experienced engineers, something that results in prolonged lead-time and increased engineering costs [16]. It has also been reported that enterprises are often unaware of the extent of in-house knowledge, whilst different sources of information regarding products, processes, and resources are not effectively coupled [15].

Knowledge-driven approaches that adopt formal knowledge models such as ontologies have been proposed as enablers of intelligent decision making. Formal knowledge models provide a shared vocabulary between distributed systems thus enabling knowledge sharing and reuse. Furthermore, they support the expression of domain specific rules that allows carrying out automated reasoning procedures on the specified knowledge [3], [24], [41], [46]. As such, they are more powerful tools than the conceptual schemas defined in industry standards (e.g. ISA-95 [39]), which focus on the structure and not the semantics of the manufacturing knowledge. Along these lines, a number of publications report on the application of ontologies in specific aspects of production, including: [17], [23], [43]–[45] on supply chain management; [2] on supply chain deployment; [11], [26], [32] on evolvable or reconfigurable production systems or processes; [36] on telecommunications product configuration; [31] on inter-enterprise collaboration; and [1] on information interoperability (introducing MSDDL, the Manufacturing Service Description Language). Although the proposed approach has been informed by such works, it is not meant to provide a generic production ontology, rather it is driven by the specific requirements of our end-users.

In terms of modeling the whole production environment in an enterprise using a formal framework, proposed approaches include ADACOR [8], [9], [28], MASON [29], and ARUM [19], [37]. ADACOR (ADaptive holonic COntrol aRchitecture) enables the formal description of a production environment using sets of autonomous, cooperative ‘holons’ corresponding to physical entities (e.g. machines or equipment) and logical entities (e.g. orders). The foundational framework for this formalism is first-order logic, which makes it hard to use in an intelligent system because reasoning in first-order logic is an undecidable problem (i.e., it is unsolvable by computational means) [7]. In addition, even in restricted cases where reasoning in first-order logic is possible, it is hard to incorporate in it data types (to reason, for example, about dates, numbers, etc.), thus making it unsuitable in practice. MASON (MANufacturing’s Semantics ONtology) is an ontology aiming to formalize the entities, processes, and resources involved in an enterprise production environment. Its formal underpinning is the OWL Web Ontology Language [18], which is founded on top of description logics. Thus, this approach is close to the ideas underpinning our proposal.

¹<http://www.disrupt-project.eu/>

However, MASON focuses mainly on structural aspects of production (e.g. the individual parts that constitute a product or production operation), neglecting behavioral aspects such as the temporal progression of production. As such it is not suitable for modeling production scheduling. In addition, the authors state that MASON has been used to automatically estimate the cost of new products; reasoning on top of the ontology is not a central consideration, which, in contrast, is important to our paper.

ARUM (Adaptive Production Management) [19], [37] is a project that aims to improve planning and control for industrial tasks in the context of aircraft manufacturing. It incorporates three main ontologies, the core ontology (aiming to describe the main aspects of production), the scene ontology (aiming to describe the state of production at a given point in time), and the events ontology (aiming to describe the entities involved in events related to production). Although this approach is closer to ours compared to the first two (i.e., ADACOR and MASON), there are differences that made it hard to adapt for our purposes. In particular, ARUM ontologies are descriptive in nature, in a sense that they provide the required means for describing concepts used by operational planners and schedulers. However, they do not aim to describe how, when, and in what way scheduling and material requirements planning takes place. In contrast, we attempt to formalize these issues and even describe formally the interactions with scheduling and material requirements planning software. Furthermore, we aim to provide a formal framework that takes into account the temporal progression of the production and describes how the progression is affected by disruptions to scheduling and material requirements planning. Finally, ARUM does not incorporate rule reasoning to encode the domain knowledge of experts.

III. ASSUMPTIONS AND MODEL OF PRODUCTION

This section describes an abstract model of production on which the proposed approach is based and describes the high level architecture of a system implementing our approach. Recall from Section I that the aim is to develop a formal framework which: (i) can effectively describe the structure of production knowledge with an emphasis on scheduling and material requirements planning aspects; (ii) can effectively model the temporal progression of production; and (iii) is practical and easy to implement.

The initial motivation for these requirements stems from our involvement in the DISRUPT project and reflects the strategic objective of the two end-user companies, namely “to improve the ability to have the materials and capability to produce without interruption”. In more detail, the first company is an automotive manufacturer with multiple production sites in several countries. Production planning and scheduling are automated to a large degree according to the capabilities of the current software configuration. However, disruptions in either inbound logistics or production, such as stock-outs or changes in the delivery of components, may affect the car units produced per person hour, something that may prove to be costly for the manufacturer. The actions to mitigate

these disruptions usually involve rescheduling the production and rearranging/reviewing orders of materials. The second company is a manufacturer of electronic boards used in several plants that produce home appliances. Due to market-driven requirements, there is great variability in the received orders from customer plants. More specifically, the quantities can be changed as well as the type of units requested. For example, a new unit is often added in the same order a few days before delivery time, or ‘urgent orders’ may appear at short notice. As a result, production at the electronics factory needs to be rescheduled frequently. A more elaborate description of the two use cases and their requirements can be found in [13], [33].

Taking into account the above, the following realistic assumptions are made to build the knowledge base. These assumptions are not meant to be generic; their main purpose is to use them to illustrate the proposed formal framework. Recall that, in the proposed framework, decision making that can help take appropriate actions has a central position, both in the level of modeling and in the level of reasoning.

- (a) A manufacturing enterprise owns multiple factories (manufacturing facilities) and each factory produces products through production lines.
- (b) Each production line involves certain machines, low level equipment (tools), and personnel.
- (c) Products are produced using parts and materials. The parts are intermediate products produced by the enterprise; the materials are ordered from external suppliers.
- (d) Products and parts, as well as materials that have been delivered, are stored in warehouses (inventories).
- (e) The production is centered around a production plan, which is set by the management and dictates how many units of each product must be produced on given dates.
- (f) To organize the production, a production schedule is derived from the production plan using scheduling software. The schedule is maintained for a fixed time period of T days. The schedule determines, for each day of this fixed time period, the allocation of resources for the production of a given family of products or parts.
- (g) To accommodate the material requirements of the production, orders of materials are arranged by specialized software for material requirements planning.
- (h) The production is supervised by a team of experts, who are responsible for ensuring that everything goes according to plan. It is also their responsibility to devise a course of action in response to any disruption that arises in production.

The role of the experts in the above model of production is summarized in Fig. 1. In this case, the experts have a central position, supervising and organizing the production with the help of material requirements planning and scheduling software (respectively, M.R. PLANNER and SCHEDULER in Fig. 1); they are also responsible for devising an appropriate action in response to an event.

The change that this paper advocates is shown in Fig. 2. In this case, the knowledge base plays a central role: it is responsible for arranging necessary actions at each point where

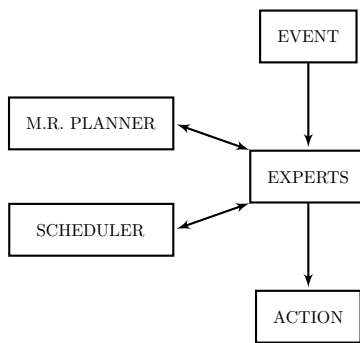


Fig. 1. Production wholly managed by a team of experts.

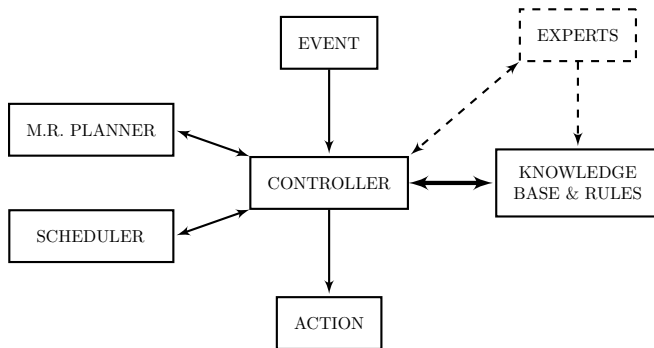


Fig. 2. Production centered around a controller/knowledge base.

a decision has to be made. The way that the knowledge base determines what needs to be done is by reasoning, based on the facts it contains and a set of rules encoding the knowledge of domain experts.

The interface between the knowledge base and other components is the CONTROLLER. The controller updates the knowledge base with the ‘state of the world’ at each point in time, and is responsible for executing any action that the knowledge base determines to be necessary. For example, the knowledge base might determine that, given the circumstances at a given point in time and based on the rules it must follow, the production of a certain product must be rescheduled. Thus, it is convenient to view the knowledge base and the controller as a unit, hence the use of a thick double arrow connecting these two components.

Experts in this model are responsible for setting and updating the rules of the knowledge base, and may not always be directly involved in production management. For this reason, the experts are depicted using dashed lines in Fig. 2. In special cases, the experts are able to override the function of the controller; this is also denoted in Fig. 2 with a double dashed arrow connecting the experts and the controller.

IV. BUILDING THE KNOWLEDGE BASE

This section provides a high-level description of a knowledge base corresponding to the description of the production environment given in Section III. For our knowledge base, we adopt the open-world assumption by standard convention. Recall that a knowledge base consists of a set of terminological axioms (called the *TBox*), a set of assertions (called the *ABox*),

and, often, in conjunction with these sets, a set of rules (involving relevant terms and individuals) in an appropriate language.

For a detailed example in RDF/XML syntax of our knowledge base we refer to <https://github.com/disrupt-project/factory>. This can be viewed in human-readable form using Protégé² or other such programs.

A. Basic concepts and roles

The first step towards building our knowledge base is the identification of the entities involved in production and their arrangement in a conceptual hierarchy. Based on the production environment described in Section III, our knowledge base features the following concepts. Note that these concepts are not meant to be a substitute for comprehensive work such as the ISA-95 standard [39] or other standardized activities [34]. These concepts are derived through our project requirements analysis and they are supposed to provide a concise, yet meaningful set to illustrate the feasibility of the proposed framework in this paper. To a large extent they can be easily mapped to the concepts of the ISA-95 standard as explained next.

Factory	Each individual of this concept corresponds to a factory, i.e., a physical, geographical, or logical grouping determined by the enterprise.
Machine	Each individual of this concept corresponds to a machine, i.e., a specific type of automated resource with specific capability.
Equipment	Each individual of this concept corresponds to a piece of equipment, i.e., a lower-level resource not defined in ISA-95, might be a tool.
ProdLine	Each individual of this concept corresponds to a production line, i.e., a series of pieces of machines dedicated to the manufacture of a specific number of products or families of products.
Personnel	Each individual of this concept corresponds to a person (worker), i.e., a human resource with specific capability.
Material	Each individual of this concept corresponds to (possibly raw) material that is used in the production of a part or product.
Part	Each individual of this concept corresponds to a part produced in a given factory. Parts are used with materials when producing products, they can be viewed as intermediate materials.
Product	Each individual of this concept corresponds to a final product.

²<https://protege.stanford.edu>

Plan	This concept is used to group individual days of the monthly production plan for any given factory.
Schedule	This concept is used to group individual days of the monthly production schedule (i.e., the allocation of resources) for each production line of any given factory.
Order	This concept is used to track daily order arrangements to enable the production of a product or part in a given factory.
Event	This concept is used to refer to events happening (in production) on individual days of a month.
Warehouse	Each individual of this concept corresponds to a warehouse.

We now introduce several roles for the formal description of various basic aspects of the production. We start with some abstract roles, which we use for a general description of the production, i.e., the arrangement of machines, equipment, and workers into production lines, and the dependencies for products:

hasProdLine	To specify the production lines in a factory.
hasMachine	To specify the machines used in a production line.
hasEquipment	To specify the equipment used in a production line.
hasPersonnel	To specify the personnel working in a production line.
requiresPart	To specify the parts required to produce a certain product.
requiresMaterial	To specify the materials required to produce a certain product.
isAvailable	To specify the availability of a given machine, piece of equipment, or person (worker).
producedIn	To specify the production line in which a product or a part is produced.

Using the above roles, the knowledge base (and, in particular, its TBox for the rest of this subsection) contains concept inclusions specifying that: a factory contains at least one production line; a production line consists of machines and also utilizes equipment and workers; a product (or part) requires parts and materials; materials, parts, and products are stored in a warehouse; each material, piece of equipment, and personnel has an ‘availability status’ which is of type boolean (i.e., it is true or false). Fig. 3 contains a diagrammatic representation of these inclusions (dashed lines represent optional roles).

The above inclusions give a qualitative description of the production. For a quantitative description, i.e., for the material requirements of products and the production schedule, we

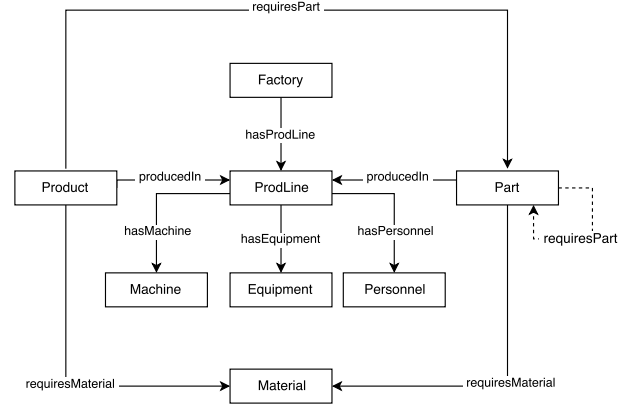


Fig. 3. Concepts and roles involved in the production of products and parts. Dashed lines indicate optional roles.

need to introduce some concrete roles. The first concrete roles that we introduce are used to record how many parts and units of materials a given product (or part) requires for its production. To this end, suppose that a company C requires for the production of its products the parts $Part_1, \dots, Part_\zeta$ and the materials Mat_1, \dots, Mat_η . We introduce, for each i ($1 \leq i \leq \zeta$) and j ($1 \leq j \leq \eta$), the following roles:

requiredUnitsOfPart _{i}	To specify the number of units of the part $Part_i$ required for the production of a given product.
requiredUnitsOfMaterial _{j}	To specify the number of units of the material Mat_j required for the production of a given product.
inventoryUnitsOf	To specify the number of units of a product, part, or material that are available in the inventory.

Based on the above roles, the knowledge base contains concept inclusions stating that each individual of the concept Product (corresponding to a product) or Part (corresponding to a part) has, for each i ($1 \leq i \leq \zeta$), a data property requiredUnitsOfPart _{i} of type integer, which we shall interpret as the number of units of the part $Part_i$ required for the production of the corresponding product or part; and, similarly, the data property requiredUnitsOfMaterial _{j} ($1 \leq j \leq \eta$) records the number of units of the material Mat_j required for the production of that product or part. Similarly, inclusions involving the roles inventoryUnitsOf state that each product, part, or material has a data property that records how many units of this product, part, or material are in the inventory. Fig. 4 contains a diagrammatic representation of these inclusions.

As described in Section III, the production is organized around a set of goals about how many units of each product should be produced in any given factory. This set of goals is set by management for every single day of a given month and encoded in the knowledge base using individuals of the

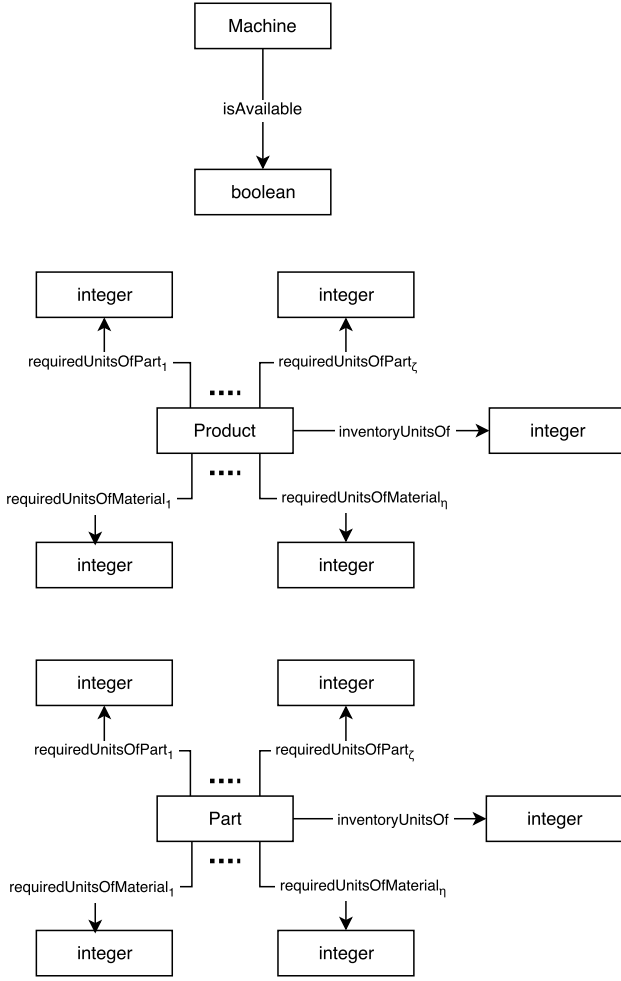


Fig. 4. Roles to describe the production in more detail.

concept Plan. Based on these goals, a production schedule (which specifies the allocation of resources within each factory for the production of a given product) arises as well as a set of order arrangements to accommodate the material requirements of the production schedule. The schedule and the order arrangements are encoded in the knowledge base using individuals of the Schedule and Order concepts, corresponding to individual days of the schedule and order arrangements, respectively.

We now introduce the roles to encode all the relevant information for the concepts Plan, Schedule, and Order. Supposing that the company produces the products $Prod_1, \dots, Prod_\xi$ and the parts $Part_1, \dots, Part_\zeta$ using the materials Mat_1, \dots, Mat_η , we introduce for each i ($1 \leq i \leq \xi$), j ($1 \leq j \leq \zeta$), and k ($1 \leq k \leq \eta$) the following roles:

- produceUnitsProd _{i} To specify the number of units of the product $Prod_i$ that the management wants to produce.
- scheduleUnitsProd _{i} To specify the number of units of the product $Prod_i$ that need to be scheduled.

- scheduleUnitsPart _{j} To specify the number of units of the part $Part_j$ that need to be scheduled.
- schedStartTimeProd _{i} To specify a start time for the production of the product $Prod_i$.
- schedEndTimeProd _{i} To specify an end time for the production of the product $Prod_i$.
- schedStartTimePart _{j} To specify a start time for the production of the part $Part_j$.
- schedEndTimePart _{j} To specify an end time for the production of the part $Part_j$.
- orderUnitsMaterial _{k} To specify the number of units of the material Mat_k that must be ordered.

Based on the above roles, the knowledge base contains concept inclusions determining the possible data properties for individual plan, schedule, and order days. In particular, it contains inclusions stating that each individual plan day may have, for each product $Prod_i$, a data property produceUnitsProd _{i} of type number, which, as discussed above, is meant to specify the number of units of the product $Prod_i$ that have to be produced. The knowledge base also contains similar inclusions for the data properties scheduleUnitsProd _{i} and scheduleUnitsPart _{j} .

Additionally, the knowledge base contains inclusions associated with days of the production schedule that define the data types of relevant data properties. Using these properties it is possible to specify a schedule to produce the product $Prod_i$ (resp. the part $Part_j$), in the form of (possibly multiple) intervals $[start-time, end-time]$, where $start-time$ is of data type time and specified by the data property schedStartTimeProd _{i} (resp. schedStartTimePart _{j}), and similarly for $end-time$ and the data property schedEndTimeProd _{i} (resp. schedEndTimePart _{j}).

Finally, the knowledge base contains inclusions stating that each individual order day may have, for each material Mat_k a data property orderUnitsMaterial _{k} of type number, which, as discussed, is meant to specify how many units of the material Mat_k must be ordered this day.

All the inclusions associated with the last set of roles are depicted in Fig. 5. The properties depicted using dashed lines are built with universal restrictions, thus they are not mandatory; but, if they exist (for a given individual), their corresponding data types must be as shown. Note that, in the second sub-figure (in the middle), the arrows labeled with ranges of indices (e.g. $1.. \xi$) are not single arrows, but represent families of arrows involving a different arrow for each of the corresponding indices (each one pointing to a box labeled 'integer').

B. Operational workflow

In the architecture we propose (recall Section III), scheduling is delegated to external scheduling software and ordering materials is delegated to software for material requirements

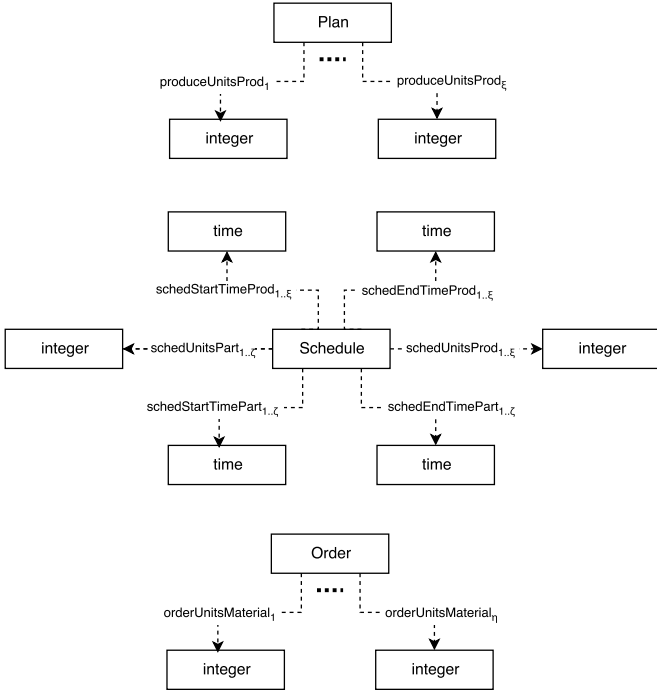


Fig. 5. Concepts and roles to describe the interaction with scheduling and material requirements planning software. Dashed lines indicate optional properties.

planning. The expected use of the last set of roles in the previous section is the following:

- The roles $\text{produceUnitsProd}_i$ are used to express the management's intention to produce a certain number of units of the corresponding product at a given set of days.
- This intention is realized as a production schedule generated by the scheduler. That the scheduler should be invoked (for the corresponding product and with the aim to schedule the given number of units) is signified by the predicates $\text{scheduleUnitsProd}_i$. Note that the scheduler may also schedule the production of parts, as needed. The scheduler, which is assumed to be external to our framework software and can be viewed as a black box, is called with the time needed to produce each product part and its dependencies. Using this input as well as information about machine configurations and their constraints, the scheduler will use appropriate scheduling algorithms to generate a plan [6], [10].
- If no conflicts arise during (b), the resulting schedule is stored in the knowledge base using the roles $\text{schedStartTimeProd}_i$, $\text{schedEndTimeProd}_i$, $\text{schedStartTimePart}_i$, and $\text{schedEndTimePart}_i$.
- Further to (b) and (c), the intention to produce a given product (and the parts that it requires) may necessitate the ordering of materials. This task is delegated to the material requirements planner, which arranges the corresponding orders.
- If no conflicts arise during (d), the resulting set of orders is stored in the knowledge base using the roles $\text{orderUnitsMaterial}_k$.

To make the knowledge base aware of possible conflicts in

the above process, we introduce the following roles:

- $\text{successSchedProd}_i$ To signify the successful generation of the production schedule for the production of Prod_i on a given day.
- $\text{successSchedPart}_j$ To signify the successful generation of the production schedule for the production of Part_j on a given day.
- $\text{successOrderMaterial}_k$ To signify a successful completion of order arrangements of Mat_k .

Based on the above roles, the knowledge base contains concept inclusions stating that each day of the schedule may have a data property signifying the successful completion of an interaction with the scheduler (to schedule the production of a product or part), and if this data property exists it is of type boolean (i.e. true for success and false for failure); similarly, each day of the concept Order (corresponding to orders) may have a data property signifying the successful completion of an interaction with a material requirements planner, and if this data property exists it is of type boolean as above. To simplify the ensuing analysis, we will assume that no failures occur when interfacing external programs, although it is not hard to incorporate possible failures in rules if one wishes to do so.

C. Individuals and assertions

As detailed in the previous section, the production within a company of a particular type (as in Section III) can be formally described in a knowledge base using various sets of concept inclusions. These inclusions describe the individual entities involved in production and how individuals of certain concepts are expected to relate to individuals of other concepts. We now start building the assertional part of such a knowledge base: we introduce the individuals themselves for each concept defined earlier and we define relations among those, thus completing the knowledge base.

We start with the introduction of individuals. Let us take the concepts in Section IV-A, one by one, for a given company of interest, C .

- **Factory:** Suppose that C owns the factories F_1, \dots, F_n (n is a positive integer); for each factory F_φ ($1 \leq \varphi \leq n$) we introduce an individual F_φ . Thus, the set

$$\mathcal{A}_0 = \{\text{Factory}(F_\varphi) \mid 1 \leq \varphi \leq n\},$$

contains the assertions that each of these individuals belongs to the concept Factory.

- **Machine:** Suppose that each factory F_φ ($1 \leq \varphi \leq n$) contains the machines $M_1^\varphi, \dots, M_{\mu_\varphi}^\varphi$. Associating to each of those machines an individual M_μ^φ ($1 \leq \mu \leq \mu_\varphi$), the set

$$\mathcal{A}_1 = \{\text{Machine}(M_\mu^\varphi) \mid 1 \leq \varphi \leq n, 1 \leq \mu \leq \mu_\varphi\}$$

contains the assertions that each individual M_μ^φ belongs to the concept MFactory_φ and, thus, corresponds to a machine (namely, M_μ^φ) at the factory F_φ .

- **Equipment:** Suppose that each factory F_φ owns the pieces of equipment $E_1^\varphi, \dots, E_{\nu_\varphi}^\varphi$ ($1 \leq \nu \leq \nu_\varphi$). As in the above, the set

$$\mathcal{A}_2 = \{\text{Equipment}(E_\nu^\varphi) \mid 1 \leq \varphi \leq n, 1 \leq \nu \leq \nu_\varphi\}$$

contains the corresponding assertions.

- **ProdLine:** Suppose that the production lines in the factory F_φ are $L_1^\varphi, \dots, L_{\lambda_\varphi}^\varphi$ ($1 \leq \lambda \leq \lambda_\varphi$). Then, the set

$$\mathcal{A}_3 = \{\text{ProdLine}(L_\lambda^\varphi) \mid 1 \leq \varphi \leq n, 1 \leq \lambda \leq \lambda_\varphi\},$$

contains the corresponding assertions.

- **Personnel:** Suppose that the personnel working in production at the factory F_φ are $P_1^\varphi, \dots, P_{\kappa_\varphi}^\varphi$ ($1 \leq \kappa \leq \kappa_\varphi$). Then, the set

$$\mathcal{A}_4 = \{\text{Personnel}(P_\kappa^\varphi) \mid 1 \leq \varphi \leq n, 1 \leq \kappa \leq \kappa_\varphi\},$$

contains the corresponding assertions.

- **Material:** Suppose that the materials used in production at any factory of the company C are Mat_1, \dots, Mat_η . Then, the set

$$\mathcal{A}_5 = \{\text{Material}(Mat_i) \mid 1 \leq i \leq \eta\},$$

contains the corresponding assertions.

- **Part:** Suppose that the parts used or produced at any factory of the company C are $Part_1, \dots, Part_\zeta$. Then, the set

$$\mathcal{A}_6 = \{\text{Part}(Part_i) \mid 1 \leq i \leq \zeta\},$$

contains the corresponding assertions.

- **Product:** Suppose that the products produced by the company C are $Prod_1, \dots, Prod_\xi$. Then, the set

$$\mathcal{A}_7 = \{\text{Product}(Prod_i) \mid 1 \leq i \leq \xi\},$$

contains the corresponding assertions.

- **Plan:** Individuals of this concept correspond to individual days within a fixed time period of T days (as described in Section III) for each given factory. These days are used to set the production goals (e.g. to produce 200 units of the product P_{25} at the factory F_3 on the 18th day of the production period) by the logistics team. Thus, the plan for each factory F_φ comprises T days $PDay_1^\varphi, \dots, PDay_T^\varphi$; to these days we associate the individuals $PDay_1^\varphi, \dots, PDay_T^\varphi$, respectively. Then the set

$$\mathcal{A}_8 = \{\text{Plan}(PDay_j^\varphi) \mid 1 \leq \varphi \leq n, 1 \leq j \leq T\},$$

contains the assertions that each $PDay_j^\varphi$ corresponds to a day (namely, $PDay_j^\varphi$) of the production plan for the factory F_φ .

- **Schedule:** Individuals of this concept correspond to individual days within a fixed time period of T days (as described in Section III) for each given factory. These are used to set the production schedule for each production day, i.e., to specify the allocation of production lines for each product within any factory. Thus, if F_φ is a factory containing the production lines $L_1^\varphi, \dots, L_{\ell_\varphi}^\varphi$, the production schedule for each production line L_j^φ ($1 \leq j \leq \ell_\varphi$) within a given factory F_φ comprises T

days $SPLineDay_1^{\varphi,j}, \dots, SPLineDay_T^{\varphi,j}$; to these days we associate the individuals $SDay_1^{\varphi,j}, \dots, SDay_T^{\varphi,j}$, respectively. Then the set

$$\mathcal{A}_9 = \{\text{Schedule}(SDay_k^{\varphi,j}) \mid 1 \leq \varphi \leq n, 1 \leq j \leq \ell_j, 1 \leq k \leq T\},$$

contains the assertions that each $SDay_k^{\varphi,j}$ corresponds to a day (namely, $SPLineDay_k^{\varphi,j}$) of the production schedule for the line L_j^φ in the factory F_φ .

- **Order:** Individuals of this concept correspond to individual days within a fixed time period of T days (as described in Section III) for each given factory. These days are used to arrange orders for the production of each given factory. Thus, the order arrangements for each factory F_φ consist of T days $ODay_1^\varphi, \dots, ODay_T^\varphi$; to these days we associate the individuals $ODay_1^\varphi, \dots, ODay_T^\varphi$, respectively. Then the set

$$\mathcal{A}_{10} = \{\text{Order}(ODay_j^\varphi) \mid 1 \leq \varphi \leq n, 1 \leq j \leq T\},$$

contains the assertions that each $ODay_j^\varphi$ corresponds to a day (namely, $SDay_j^\varphi$) of order arrangements regarding the factory F_φ .

- **Event:** Individuals of this concept correspond to individual days within a fixed time period of T days (as described in Section III) for each given factory. These days are used to specify events in production for each given factory. Thus, the production for each factory F_φ consists of T days $EDay_1^\varphi, \dots, EDay_T^\varphi$ on each of which multiple events can occur; to these days we associate the individuals $EDay_1^\varphi, \dots, EDay_T^\varphi$, respectively. Then the set

$$\mathcal{A}_{11} = \{\text{Event}(EDay_j^\varphi) \mid 1 \leq \varphi \leq n, 1 \leq j \leq T\},$$

contains the assertions that each $EDay_j^\varphi$ corresponds to a day (namely, $EDay_j^\varphi$) on which multiple events can occur regarding the production in the factory F_φ .

- **Warehouse:** Suppose that C owns the warehouses W_1, \dots, W_q (q is a positive integer); for each warehouse W_i ($1 \leq i \leq q$) we introduce an individual W_i . Thus, the set

$$\mathcal{A}_{12} = \{\text{Warehouse}(W_i) \mid 1 \leq i \leq q\},$$

contains the assertions that each of these individuals belongs to the concept Warehouse.

Let us now move to assertions about roles. For the rest of the section, the ranges of the subscripts are as above. We first turn our attention to roles associated with Fig. 3. Let \mathcal{A}_{13} be the ABox comprising the following assertions:

- **hasProdLine**($F_\varphi, L_\lambda^\varphi$), if the factory F_φ contains the production line L_λ^φ ;
- **hasMachine**($L_\lambda^\varphi, M_\mu^\varphi$), if the production line L_λ^φ uses the machine M_μ^φ ;
- **hasEquipment**($L_\lambda^\varphi, E_\nu^\varphi$), if the production line L_λ^φ involves the piece of equipment E_ν^φ ;
- **hasPersonnel**($L_\lambda^\varphi, P_\kappa^\varphi$), if P_κ^φ works at the production line L_λ^φ ;
- **requiresPart**($Prod_i, Part_j$), if $Prod_i$ requires $Part_j$ for its production;

- $\text{requiresMaterial}(\text{Prod}_i, \text{Mat}_j)$, if Prod_i requires Mat_j for its production;
- $\text{isAvailable}(M_\mu^\varphi, \text{true})$, if the machine M_μ^φ is available, $\text{isAvailable}(M_\mu^\varphi, \text{false})$ if not;
- $\text{producedIn}(\text{Prod}_i, L_\lambda^\varphi)$, if Prod_i is produced in the production line L_λ^φ ; and
- $\text{producedIn}(\text{Part}_i, L_\lambda^\varphi)$, if Part_i is produced in the production line L_λ^φ .

For the roles associated with Fig. 4, let \mathcal{A}_{14} be the ABox comprising the following assertions:

- $\text{requiredUnitsOfPart}_i(\text{Prod}_k, u)$ if Prod_k requires u units of Part_i for its production;
- $\text{requiredUnitsOfPart}_i(\text{Part}_k, u)$ if Part_k requires u units of Part_i for its production;
- $\text{requiredUnitsOfMaterial}_j(\text{Prod}_k, u)$ if Prod_k requires u units of Mat_j for its production;
- $\text{requiredUnitsOfMaterial}_j(\text{Part}_k, u)$ if Part_k requires u units of Mat_j for its production;
- $\text{inventoryUnitsOf}(\text{Prod}_i, u)$ (respectively $\text{inventoryUnitsOf}(\text{Part}_j, u)$), if there are u units of Prod_i (respectively Part_j) in the inventory.

Finally, for roles $\text{produceUnitsProd}_i$ associated with Fig. 5, let \mathcal{A}_{15} be the ABox comprising the following assertions: $\text{produceUnitsProd}_i(\text{PDay}_t^\varphi, u)$, if the management wants to produce u units of Prod_i on day t of the production month at factory F_φ . Recall, as discussed in Section IV-B, that the rest of the roles shown in Fig. 5 are not to be used by humans: they are set by the controller as a result of its interaction with scheduling and material requirements planning software. The same applies to the roles appearing in Section IV-B.) Then, the ABox $\mathcal{A} = \mathcal{A}_0 \cup \dots \cup \mathcal{A}_{15}$ contains the assertions describing the whole production (with respect to our model; recall Section III).

V. A PRODUCTION MANAGEMENT SCENARIO

In the previous sections we built the necessary formal framework to define rules for production management. We now develop a set of rules on top of this framework based on realistic use cases. The rules are written in the SWRL [22] language, which is very commonly used with description logics. To save space, we prefix SWRL built-in functions with ‘:’ instead of ‘swrlb:’. We also remark that there are various other languages for the specification of rules, as, for example, SPIN³ to represent SPARQL rules.

Let \mathcal{K} be the knowledge base comprising the TBox described in Section IV-A and the ABox $\mathcal{A}_0 \cup \dots \cup \mathcal{A}_{15}$ as described in Section IV-C. All the following rules should be used in conjunction with \mathcal{K} . We remark that the rules written below are not single rules, but families of rules for different values of their indices (the ranges of the indices are as in Section IV-C). The process of writing these rules is tedious, but, at the same time, demonstrates the large number of cases and interactions that human experts need to consider to make decisions and, thus, the difficulty of this task.

Basic production management

When the management has set a plan to produce x units of Prod_i ($1 \leq i \leq \xi$) on day t ($1 \leq t \leq T$) of the fixed time period of T days (recall Section III) at the factory F_φ ($1 \leq \varphi \leq n$), the production of this product at the appropriate production line L_λ^φ ($1 \leq \lambda \leq \lambda_i$) must be scheduled to meet this goal:

$$\begin{aligned} & \text{produceUnitsProd}_i(\text{PDay}_t^\varphi, ?x) \wedge \text{producedIn}(\text{Prod}_i, L_\lambda^\varphi) \\ & \rightarrow \text{scheduleUnitsProd}_i(\text{SDay}_t^{\varphi, \lambda}, ?x). \end{aligned}$$

Further, the orders of necessary materials must be arranged. That is, for each material Mat_k ($1 \leq k \leq \eta$), if the production of Prod_i requires Mat_k , the required number of units for that material must be ordered:

$$\begin{aligned} & \text{produceUnitsProd}_i(\text{PDay}_t^\varphi, ?x) \\ & \wedge \text{requiresMaterial}(\text{Prod}_i, \text{Mat}_k) \\ & \wedge \text{requiredUnitsOfMaterial}_k(\text{Prod}_i, ?u) \\ & \wedge \text{:multiply}(?x \text{Times} U, ?x, ?u) \\ & \rightarrow \text{orderUnitsMaterial}_k(\text{ODay}_t^\varphi, ?x \text{Times} U). \end{aligned}$$

The production of Prod_i may also require the production of parts. Thus, the production of the necessary parts must be arranged. That is, for each part Part_j ($1 \leq j \leq \zeta$), if Prod_i requires Part_j , then Part_j must also be produced:

$$\begin{aligned} & \text{produceUnitsProd}_i(\text{PDay}_t^\varphi, ?x) \\ & \wedge \text{requiresPart}(\text{Prod}_i, \text{Part}_j) \\ & \wedge \text{requiredUnitsOfPart}_j(\text{Prod}_i, ?u) \\ & \wedge \text{:multiply}(?x \text{Times} U, ?x, ?u) \\ & \rightarrow \text{produceUnitsPart}_j(\text{PDay}_t^\varphi, ?x \text{Times} U). \end{aligned}$$

Just like with products, each part Part_j ($1 \leq j \leq \zeta$) that is to be produced must be scheduled for production at the appropriate production line L_λ^φ :

$$\begin{aligned} & \text{produceUnitsPart}_j(\text{PDay}_t^\varphi, ?x) \wedge \text{producedIn}(\text{Part}_j, L_\lambda^\varphi) \\ & \rightarrow \text{scheduleUnitsPart}_j(\text{SDay}_t^{\varphi, \lambda}, ?x). \end{aligned}$$

In addition, the production of a part may require certain materials to be ordered and other parts (recursively) to be produced. For each material Mat_k ($1 \leq k \leq \eta$), if the production of Part_j requires Mat_k , the required number of units for that material must be ordered:

$$\begin{aligned} & \text{produceUnitsPart}_j(\text{PDay}_t^\varphi, ?x) \\ & \wedge \text{requiresMaterial}(\text{Part}_j, \text{Mat}_k) \\ & \wedge \text{requiredUnitsOfMaterial}_k(\text{Part}_j, ?u) \\ & \wedge \text{:multiply}(?x \text{Times} U, ?x, ?u) \\ & \rightarrow \text{orderUnitsMaterial}_k(\text{ODay}_t^\varphi, ?x \text{Times} U). \end{aligned}$$

For each part $\text{Part}_{j'}$ ($1 \leq j' \leq \zeta$), if Part_j requires $\text{Part}_{j'}$, then $\text{Part}_{j'}$ must also be produced:

$$\begin{aligned} & \text{produceUnitsPart}_j(\text{PDay}_t^\varphi, ?x) \\ & \wedge \text{requiresPart}(\text{Part}_j, \text{Part}_{j'}) \end{aligned}$$

³<http://spinrdf.org/>

$$\begin{aligned}
& \wedge \text{requiredUnitsOfPart}_{j'}(\text{Part}_{j'}, ?u) \\
& \wedge : \text{multiply}(?x\text{Times}U, ?x, ?u) \\
& \rightarrow \text{produceUnitsPart}_{j'}(\text{PDay}_t^\varphi, ?x\text{Times}U).
\end{aligned}$$

Note that we have now handled (recursively) all possible requirements for the production.

Disruptions in inbound logistics

An important source of disruptions is related to inbound logistics. These are handled by a (dedicated to this task) human expert, whose responsibility is to ensure the timely delivery of components to the lineside. This person's work is to update delivery plans (up to several times per day) in case of events occurring in supplier production and transportation. The most common events related to a supplier's production are delays as well as availability, quantity, and quality issues. When the production of a certain component is delayed, the component is not available (either at all or at the required quantity), or its production quality is not as expected, the expert picks an appropriate supplier as a replacement (using the SAP® material requirements planner) and reschedules the production if necessary. Common events related to transportation include vehicle breakdown, delay, and errors in delivery. Based on experience and with the help of the appropriate software (as above), the expert selects another carrier and/or reschedules the production.

To automate the above decisions, we assume that an integrated cyber-physical system [27] provides data to track delays as well as availability, quality, and/or quantity issues on behalf of each supplier, and, in addition, movement of track delivery vehicles and their deliveries. We assume that the knowledge base has access to such data through certain predefined roles. For example, in the following set of rules we use the roles $\text{supplierDelayMat}_k$ ($1 \leq k \leq \eta$), which we assume track (through access of collected data) delays on a supplier's production of Mat_k ($1 \leq k \leq \eta$).

Suppose that Prod_i ($1 \leq i \leq \xi$) requires material Mat_k ($1 \leq k \leq \eta$); then the rule

$$\begin{aligned}
& \text{produceUnitsProd}_i(\text{PDay}_t^\varphi, ?u) \\
& \wedge \text{requiresMaterial}(\text{Prod}_i, \text{Mat}_k) \\
& \wedge \text{supplierDelayMat}_k(\text{ODay}_t^\varphi, ?d) \\
& \rightarrow \text{scheduleUnitsProd}_i(\text{SDay}_t^{\varphi, \lambda}, ?u)
\end{aligned}$$

states that if there is a delay on behalf of the supplier of a required material for the production of a given product, then the production of that product must be rescheduled. This includes possibly ordering the material from a different supplier if necessary, something decided by the material requirements planner. And, similarly, for parts:

$$\begin{aligned}
& \text{produceUnitsPart}_j(\text{PDay}_t^\varphi, ?u) \\
& \wedge \text{requiresMaterial}(\text{Part}_j, \text{Mat}_k) \\
& \wedge \text{supplierDelayMat}_k(\text{ODay}_t^\varphi, ?d) \\
& \rightarrow \text{scheduleUnitsProd}_i(\text{SDay}_t^{\varphi, \lambda}, ?u).
\end{aligned}$$

In an analogous manner, one can write similar rules to handle possible availability, quantity, or quality issues on behalf of the supplier, as well as transportation delays.

Disruptions in production scheduling

Another important source of disruptions is related to production scheduling. Such disruptions are handled by a human expert (different to the one handling disruptions in inbound logistics), who is responsible for the production scheduling of a whole plant. This person's job is to replan (possibly several times per day) the sequence of vehicles going through body-in-white, painting, and assembly. The sequence break ('*rottura sequenza*') is motivated by process events such as machine breakdown, unscheduled maintenance, software problems, etc.

According to the above, suppose that Prod_i ($1 \leq i \leq \xi$) is produced at production line L_λ^φ of the factory F_φ , and that machine M_μ^φ becomes unavailable. Then, the rules

$$\begin{aligned}
& \text{produceUnitsProd}_i(\text{PDay}_t^\varphi, ?u) \wedge \text{producedIn}(L_\lambda^\varphi, \text{Prod}_i) \\
& \wedge \text{hasMachine}(L_\lambda^\varphi, M_\mu^\varphi) \wedge \text{isAvailable}(M_\mu^\varphi, \text{false}) \\
& \rightarrow \text{scheduleUnitsProd}_i(\text{SDay}_t^{\varphi, \lambda}, ?u)
\end{aligned}$$

and

$$\begin{aligned}
& \text{produceUnitsPart}_i(\text{PDay}_t^\varphi, ?u) \wedge \text{producedIn}(L_\lambda^\varphi, \text{Part}_i) \\
& \wedge \text{hasMachine}(L_\lambda^\varphi, M_\mu^\varphi) \wedge \text{isAvailable}(M_\mu^\varphi, \text{false}) \\
& \rightarrow \text{scheduleUnitsPart}_i(\text{SDay}_t^{\varphi, \lambda}, ?u)
\end{aligned}$$

state that if a machine becomes unavailable, the products and parts produced at the production line to which that machine belongs must be rescheduled. We assume that the scheduler can take the availability of machines into account by accessing the isAvailable property. Similar rules can be written to handle additional disruptions like the ones mentioned above.

VI. CONCLUSIONS AND FUTURE WORK

The paper has demonstrated that it is possible to develop a formal framework for describing the organizational structure of production in modern manufacturing enterprises, with an emphasis on production scheduling and material requirements planning, and which can lay the foundations for an intelligent system that encodes the domain knowledge of experts as sets of rules. Such a system can be an invaluable tool in helping humans manage the complexities of decision making in modern production scenarios. The rules provided in this paper only cover a small spectrum of the potential of such a framework. The underlying concepts may form a basis to investigate the possibility of extending coverage to other production management activities. Although a complete set of rules for a modern manufacturing enterprise could easily consist of thousands of rules, it may be the case that many of these rules would be instances of a much smaller set of distinct classes of rules.

One the other hand, it is recognized that building a system based on the proposed framework may have a significant initial cost. Some of the framework's expressiveness may have to be compromised to reduce the overall complexity of reasoning;

for example, a fixed time period helps dealing with temporal properties without resorting to temporal logics. In addition, determining what aspects of production management may have to be given priority when building the system's formalism may, in general, be challenging. The counter-argument is that for specific scenarios, as the ones discussed in the paper, the proposed framework may empower decision makers when facing complex decision making processes. In this direction, an important extension of this research would be to incorporate reasoning under uncertainty. Indeed, the framework presented in this paper can only handle definitive knowledge. However, in modern production, one cannot always make definitive statements about the state of a given system. As humans have difficulties with uncertain reasoning, the proposed system could be adapted to include rules capturing uncertain knowledge. As mentioned in Section I, there are various interesting results and research directions regarding, for example, probabilistic description logics, and, in this connection, our research could provide a bridge towards production management under uncertainty.

REFERENCES

- [1] Farhad Ameri and Debasish Dutta. An upper ontology for manufacturing service description. In *ASME 2006 international design engineering technical conferences and computers and information in engineering conference*, pages 651–661. American Society of Mechanical Engineers, 2006.
- [2] Farhad Ameri and Lalit Patil. Digital manufacturing market: A semantic web-based framework for agile supply chain deployment. *Journal of Intelligent Manufacturing*, 23:1817–1832, 2012.
- [3] Farhad Ameri, Colin Urbanovsky, and Christian McArthur. A systematic approach to engineering knowledge organization and modeling. In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 947–958. American Society of Mechanical Engineers, 2012.
- [4] Franz Baader, Maurizio Lenzerini, and Riccardo Rosati. *The description logic handbook: theory, implementation, and applications*. 2003.
- [5] Armin Biere, Marijn Heule, and Hans van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009.
- [6] Jacek Blazewicz, Klaus H. Ecker, Erwin Pesch, Günter Schmidt, and Jan Weglarz. *Scheduling computer and manufacturing processes*. Springer science & Business media, 2013.
- [7] Egon Börger, Erich Grädel, and Yuri Gurevich. *The classical decision problem*. Springer Science & Business Media, 2001.
- [8] Stefano Borgo and Paulo Leitão. The role of foundational ontologies in manufacturing domain applications. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 670–688, 2004.
- [9] Stefano Borgo and Paulo Leitão. Foundations for a core ontology of manufacturing. In Raj Sharman, Rajiv Kishore, and Ram Ramesh, editors, *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems*, pages 751–775. Springer US, Boston, MA, 2007.
- [10] Peter Brucker. *Scheduling algorithms*, volume 3. Springer, 2007.
- [11] Gonçalo Candido, Armando W. Colombo, José Barata, and François Jammes. Service-oriented infrastructure to support the deployment of evolvable production systems. *IEEE Transactions on Industrial Informatics*, 7(4):759–767, Nov. 2011.
- [12] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014.
- [13] Pavlos Eirinakis, Jorge Buenabad-Chavez, Rosanna Fornasiero, Haluk Gokmen, Julien-Etienne Mascolo, Ioannis Mourtos, Sven Spieckermann, Vasilios Tountopoulos, Frank Werner, and Robert Woitsch. A proposal of decentralised architecture for optimised operations in manufacturing ecosystem collaboration. In Luis M. Camarinha-Matos, Hamideh Afsarmanesh, and Rosanna Fornasiero, editors, *Collaboration in a Data-Rich World*, pages 128–137. Springer International Publishing, 2017.
- [14] Soumaya El Kadiri, Bernard Grabot, Klaus-Dieter Thoben, Karl Hribernik, Christos Emmanouilidis, Gregor Von Cieminski, and Dimitris Kiritsis. Current trends on ICT technologies for enterprise information systems. *Computers in Industry*, 79:14–33, 2016.
- [15] Borja Ramis Ferrer, Bilal Ahmad, Andrei Lobov, Daniel Alexandre Vera, José Luis Martínez Lastra, and Robert Harrison. An approach for knowledge-driven product, process and resource mappings for assembly automation. In *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*, pages 1104–1109. IEEE, 2015.
- [16] Borja Ramis Ferrer, Sergii Iarovy, Andrei Lobov, and José L. Martínez Lastra. Towards processing and reasoning streams of events in knowledge-driven manufacturing execution systems. In *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*, pages 1075–1080. IEEE, 2015.
- [17] Guido L. Geerts and Daniel E. O’Leary. A supply chain of things: The EAGLET ontology for highly visible supply chains. *Decision Support Systems*, 63:3–22, 2014.
- [18] W3C Owl Working Group et al. OWL 2 web ontology language document overview. 2009.
- [19] Ondřej Hrcuba and Pavel Vrba. Ontologies for flexible production systems. In *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*, pages 1–8. IEEE, 2015.
- [20] Mario Hermann, Tobias Pentek, and Boris Otto. Design principles for industrie 4.0 scenarios. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 3928–3937. IEEE, 2016.
- [21] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. Justification oriented proofs in OWL. In *Lecture Notes in Computer Science*, volume 6496, pages 354–369, 2010.
- [22] Ian Horrocks and Peter F. Patel-Schneider. A Proposal for an OWL Rules Language. In *Proceedings of the 13th International Conference on World Wide Web, WWW ’04*, pages 723–731, New York, NY, USA, 2004. ACM.
- [23] Michael N. Huhns, Larry M. Stephens, and Nenad Ivezic. Automating supply-chain management. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 3, AAMAS ’02*, pages 1017–1024, New York, NY, USA, 2002. ACM.
- [24] Sergii Iarovy, Wael M. Mohammed, Andrei Lobov, Borja Ramis Ferrer, and Jose L. Martinez Lastra. Cyber-physical systems for open-knowledge-driven manufacturing execution systems. *Proceedings of the IEEE*, 104(5):1142–1154, 2016.
- [25] Evangelia Kavakli, Jorge Buenabad-Chávez, Vasilios Tountopoulos, Pericles Loucopoulos, and Rizos Sakellariou. An architecture for disruption management in smart manufacturing. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 279–281, June 2018.
- [26] Jose L. Martinez Lastra and Ivan M. Delamer. Semantic web services in factory automation: Fundamental insights and research roadmap. *IEEE Transactions on Industrial Informatics*, 2:1–11, 2006.
- [27] Paulo Leitão, Armando Walter Colombo, and Stamatis Karnouskos. Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Computers in Industry*, 81:11–25, 2016.
- [28] Paulo Leitão and Francisco Restivo. ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, 57(2):121–130, 2006.
- [29] Séverin Lemaignan, Ali Siadat, Jean Yves Dantan, and Anatoli Semenenko. MASON: A proposal for an ontology of manufacturing domain. In *Proceedings DIS 2006: IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, pages 195–200, 2006.
- [30] Pedro Lima-Monteiro, Mafalda Parreira-Rocha, André Dionisio Rocha, and Jose Barata Oliveira. Big Data Analysis to Ease Interconnectivity in Industry 4.0 – A Smart Factory Perspective. In *Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 237–245. Springer, 2017.
- [31] H. K. Lin and J. A. Harding. A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. *Computers in Industry*, 58:428–437, 2007.
- [32] Niels Lohse, Hitendra Hirani, and Svetan Ratchev. Equipment ontology for modular reconfigurable assembly systems. *International Journal of Flexible Manufacturing Systems*, 17(4):301–314, Oct 2006.
- [33] P. Loucopoulos and E. Kavakli. Analysis of requirements for a cyber physical production system in the automotive industry. In *AMCIS 2017, Boston, USA*, 2017.
- [34] Arndt Lüder, Nicole Schmidt, and Rainer Drath. Standardized information exchange within production system engineering. In Stefan Biffl, Arndt Lüder, and Detlef Gerhard, editors, *Multi-Disciplinary*

Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects, pages 235–257. Springer International Publishing, 2017.

- [35] Thomas Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172:852–883, 2008.
- [36] Deborah L. McGuinness and Jon R. Wright. An industrial-strength description logic-based configurator platform. *IEEE Intelligent Systems and Their Applications*, 13:69–77, 1998.
- [37] Despina T. Meridou, Andreas P. Kapsalis, Maria-Eleftheria Ch. Papadopoulou, Emmanouil G. Karamanis, Charalampos Z. Patrikakis, Iakovos S. Venieris, and Dimitra-Theodora I. Kaklamani. An Ontology-Based Smart Production Management System. *IT Professional*, 17:36–46, 2015.
- [38] David Romero and François Vernadat. Enterprise information systems state of the art: past, present and future trends. *Computers in Industry*, 79:3–13, 2016.
- [39] Bianca Scholten. *The Road to Integration. A Guide to Applying the ISA-95 Standard in Manufacturing*. Instrumentation, Systems and Automation Society, 2007.
- [40] Fei Tao, Ying Cheng, Li Da Xu, Lin Zhang, and Bo Hu Li. CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system. *IEEE Transactions on Industrial Informatics*, 10(2):1435–1442, 2014.
- [41] Zahid Usman, R.I.M. Young, Nitishal Chungoora, Claire Palmer, Keith Case, and J.A. Harding. Towards a formal manufacturing reference ontology. *International Journal of Production Research*, 51(22):6553–6572, 2013.
- [42] Jan Vom Broecke and Michael Rosemann. *Handbook on business process management*. Springer, 2010.
- [43] Xiaohuan Wang, T. N. Wong, and Zhi Ping Fan. Ontology-based supply chain decision support for steel manufacturers in China. *Expert Systems with Applications*, 40:7519–7533, 2013.
- [44] Yan Ye, Dong Yang, Zhibin Jiang, and Lixin Tong. An ontology-based architecture for implementing semantic integration of supply chain management. *International Journal of Computer Integrated Manufacturing*, 21:1–18, 2008.
- [45] Yan Ye, Dong Yang, Zhibin Jiang, and Lixin Tong. Ontology-based semantic models for supply chain management. *International Journal of Advanced Manufacturing Technology*, 37:1250–1260, 2008.
- [46] Yingzhong Zhang, Xiaofang Luo, Hong Zhang, and John W. Sutherland. A knowledge representation for unit manufacturing processes. *The International Journal of Advanced Manufacturing Technology*, 73(5-8):1011–1031, 2014.



Georgios Kourtis holds a PhD in Computer Science from the University of Manchester (UK), an MSc in Logic, Algorithms, and the Theory of Computation from the University of Athens (Greece) and the National Technical University of Athens (Greece), and a BSc in Computer Science from the University of Piraeus (Greece). His research interests are broadly on the applications of logic in computer science, and, in particular, on the complexity of reasoning in various decidable logics (e.g. fragments of first-order logic, description logics, temporal logics) and

their connections to knowledge representation, database theory, and formal verification.



Evangelia Kavakli is an Associate Professor at the Department of Cultural Technology and Communication of the University of the Aegean (Greece) and a Research Associate in the School of Computer Science of the University of Manchester (UK). She obtained her PhD in Computation from the University of Manchester, Institute of Science and Technology in 1999. She is in charge of the Cultural Informatics Laboratory of the University of the Aegean. Her research on the topics of goal-oriented requirements engineering, enterprise knowledge modelling, information systems privacy and cultural informatics, has been supported in the context of national and EU funded projects. Her current research focuses on requirements engineering for Big Data applications and the design of socio-cyber-physical systems. She has published over 60 peer-reviewed papers in reputed international journals and conferences and edited books. She is member of the editorial board of the Requirements Engineering Journal and has served in the program committee of a substantial number of international conferences in the field of information systems.



Rizos Sakellariou obtained his PhD in Computer Science from the Victoria University of Manchester in 1997. Following positions with Rice University and the University of Cyprus he has been an academic member of staff of the University of Manchester since 2000, currently Professor of Computer Science and leader of a laboratory that carries out research in High-Performance, Parallel and Distributed Software Systems. He has published over 150 research papers, his research has been supported by several UK and EU projects and has

served on the Program Committee of over 160 conferences and workshops. He values collaboration and a strong work ethic.