Improved Domain Modeling for Realistic Automated Planning and Scheduling in Discrete Manufacturing

Antje Rogalla[1]
[1]Institute Industrial IT
OWL University of Applied Sciences
Lemgo, Germany
antje.rogalla@stud.hs-owl.de

Alexander Fay[2]
[2]Institute of Automation Technology
Helmut-Schmidt-University
Hamburg, Germany
alexander.fay@hsu-hh.de

Oliver Niggemann[1,3]
[3]Fraunhofer IOSB-INA
Application Center Industrial Automation
Lemgo, Germany
oliver.niggemann@iosb-ina.fraunhofer.de

***Abstract*—Current production planning and scheduling systems in automation do not meet the requirements of modern individualized production. Today's, static production processes impede customized manufacturing and small-scale production. A new way of thinking towards a dynamic control is required. This paper focuses on automated integrated process planning and scheduling on control level in discrete manufacturing. Existing algorithms in artificial intelligence planning are applied to solve process planning and scheduling problems. The challenge is to model the manufacturing system and products in a way that automated planners can generate efficiently process plans and schedules. Hence, based on a general classification of operations, different modeling options with regard to a successful automated process planning and scheduling are discussed. As a result, a domain modeling approach for discrete manufacturing is presented.**

I. Motivation

Smart factories are the vision of Industrie 4.0 [1]. Cyber-Physical Production Systems (CPPSs) are technical enablers and a key technology for smart factories [2]. Main features of CPPSs are *adaptability* and *self-organization*[3]: They are expected to be flexible, reconfigurable and adaptable and react autonomously to customer demands and a changing manufacturing environment.
Customized manufacturing and small-scale production are the trend in modern manufacturing industry [1]. The increasing complexity of products and manufacturing systems will result in a change in the manufacturing planning paradigm by introducing new methods and process technologies over the coming decade [2].

Today, during factory planning engineers first model production processes e.g. with the formalised process descriptions VDI/VDE 3682 [4], then implement software for the programmable controllers e.g. using IEC 61131-3, one of the most successful global standards for industrial control software [5]. Production processes of manufacturing systems are manually implemented in static control systems [6]. Any change to processes requires complex and time-consuming manual software rewriting of control applications [7].
Conventional automation is hierarchically structured. Scheduling of product orders is part of the *Manufacturing Execution Systems (MES)* [8] and superordinate to manufacturing control. Traditionally, manufacturing process planning and scheduling are two separated tasks. Manufacturing process planning determines how a product will be manufactured, while manufacturing scheduling is the task to assign "manufacturing resources over time to the set of manufacturing processes in the process plan" [9]. Modern individualized production call for a new way of thinking in automation related to manufacturing control: An automated integrated process planning and scheduling on control level to establish a dynamic control software. Scheduling is no superior task of the MES, but integrated into the control system. Process planning and scheduling merge and based on customer-specific orders, production processes are planed and scheduled automatically.

In CPPS with a modular design production modules can be replaced or added in response to new product requirements. Particularly for these systems on software level a dynamic control is required to guarantee the adaptability. The main idea of this paper is to conceptualize a new planning system for modular CPPSs. Existing methods and standard algorithms in artificial intelligence (AI) planning are applied to enable an automated integrated process planning and scheduling in automation. The modeling of the manufacturing planning problem represents a key challenge:
Essential prerequisite for the successful application of existing algorithms in AI planning is the modeling of the CPPS, the individual modules and their skills, and the ordered products as manufacturing planning problem. This has to be done in a way that the planning algorithms can solve this manufacturing planning problem and generate automatically and efficiently process plans/schedules. To open AI planning to automation, an appropriate modeling concept to model manufacturing systems and orders need to be developed.

Focus of this paper is the development of an appropriate modeling concept initially for discrete manufacturing. With this modeling concept the use of AI planning methods on control level becomes possible. Then new product requirements or changes on the technical system require no manual adaption of production processes. AI planning on control level enables an automated generation of process plans/schedules in consideration of dynamic manufacturing environments.

Compared to classical benchmarks in domain independent planning of AI planning, e.g. the blocksworld planning problem [10], discrete manufacturing planning problems are sig-

978-1-5386-7108-5/18/$31.00 ©2018 IEEE

nificantly more complex: Planning domains are large heterogeneous systems consisting of different modules, e.g. transport and production modules. A range of dependencies, e.g. between modules, exists and needs to be appropriately modeled in order to prevent the generation of wrong production processes. Concerning this matter, the following research questions need to be answered:

Research question 1. CPPSs consist of different modules. These modules are able to execute operations. In order to reach final products during operations, attributes of input products change. A rotary plate executes a transport operation, e.g. products are rotated and change their spatial position. A screw module executes a production operation, e.g. a lid is screwed on a bottle.
Toward a general modeling concept it is necessary to classify operations according to how they change attributes on products. Is such a classification possible?

Research question 2. (Based on RQ 1) is it possible to develop a general modeling concept of discrete manufacturing planning problems, such that existing algorithms in AI Planning are applicable to plan and schedule on control level? To achieve this, it has to figure out which way of modeling is suitable for a successful automated planning and scheduling of manufacturing planning problems, i.a.: How can discrete manufacturing products or executable operations of modules be modeled?

Objective of this paper is to build a bridge between the two areas of research *automation* on the one site and *AI planning* on the other site and to present a domain modeling approach for realistic automated planning and scheduling in discrete manufacturing.

II. State of the Art

A successful automated process planning and scheduling depend on, first, an appropriate modeling of the manufacturing planning problem. Second, an efficient planning: The application of a planning software, which is able to solve the planning problem and generates efficiently process plans/schedules. Both modeling and planning are closely linked and presuppose one another. The use of an automated planner requires the modeling of the planning problem in a way that the automated planner is able to understand the modeling and to solve the problem. Furthermore, appropriate modeling always depends on the selection of the automated planner.

In the last decade the planning language **Planning Domain Definition Language (PDDL)** has established as standard formalisme for classical planning [11]. The language was introduced 1998 on the first **International Planning Competition (IPC)**. Since then the language was continuously evolved and expanded to include further modeling options. Today the versions 1.2 [11], 2.1 [12], 2.2 [13], 3.0 [14] and 3.1 [15] of PDDL exist. PDDL 2.1 enables in addition to PDDL 1.2 the modeling of i.a *durative actions* (and in a limited way *continuous actions*) and *plan-metrics*. In PDDL 2.2 *derived predicates* and *timed initial literals* were introduced. With version 3.0 i.a. the modeling of *preferences* is possible. *Object-fluents* are supplemented in Version 3.1.
Furthermore, based on PDDL e.g. the language PDDL+ [16] was developed. PDDL+ is an extension of PDDL 2.1. and improves i.a. the modeling of *continuous actions*. Further expansions of PDDL see e.g. [17] exist, but they do not support the modeling of time. With regard to manufacturing planning problems one significant precondition is that the modeling language enables the model processing times. Otherwise scheduling of operations and production processes is not possible.

Since 1998 periodically (about every two years) the IPC is organized [18], [19]. On the basis of a set of benchmarks participating automated planners are tested. The participating planners not need to support all requirements of PDDL. For some years now, e.g. temporal planners participating on the temporal track of the IPC need to support a subset of PDDL 2.2. The planner have to support at least the requirements **STRIPS** see e.g. [20], *durative actions* and *action costs*. The last temporal track was organized 2014 within the deterministic part of the IPC 2014. Criticism of the IPC 2014 was i.a. that few participating planners could deal with preferences or temporal models [19].
To the best of the author's knowledge, until now, no planners supporting all fragments of PDDL 3.1. were developed.

Although PDDL and its extensions were developed to solve real world problems, the application of automated planning and scheduling in manufacturing is rare. An application of automated process planning for CNC machining is discussed in [21]. In [22] the greenhouse logistics management is introduced as classical planning problem. The applications focus on process planning and not on scheduling. They are simplified planning problems and do not solve real world manufacturing planning problems. The reasons for this can be found in, first, PDDL as language is still not sufficiently expressive to model complex manufacturing planning problems. Second, existing automated planners cannot handle the complexity of manufacturing planning problems.
Regarding logistics first applications of *AI planning* exists to plan and control industrial robots. In 2012 the **RoboCup Logistics League (RCLL)** was founded to tackle the problem of production logistics [23]. RCLL is an annual event. On a defined field different production modules (sponsored by Festo) are located. Two teams of up to three robots compete simultaneously. The challenge of the robots is "to plan, execute, and optimise the material flow in a smart factory scenario and deliver products according to dynamic orders." [23] Note that the focus of RCLL is on planning robots according to in-factory logistics. The goal is not a general production planning and not an efficient automated planning and scheduling of product orders.
A general approach of automated planning and scheduling in production planning does not yet exist in automation.

III. A CLASSIFICATION OF OPERATIONS IN DISCRETE MANUFACTURING

Discrete manufacturing is “the production of distinct items such as automobiles, appliances or computers.” [24] Discrete manufacturing input, intermediate and end products are clearly identifiable. According to the definition discrete manufacturing covers a broad spectrum of products and manufacturing systems.

A successful automated discrete manufacturing process planning and scheduling assumes the description of the manufacturing system, the individual modules and their skills, and the specification of ordered products. To open automated process planning and scheduling to discrete manufacturing a general classification of discrete manufacturing processes is needed. A suitable criterion to classify the different processes on a product during the production process is the change of attributes. For example, during a transport operation not the product itself, but its spatial position changes.

In this section first the description of products is formalised, then production process and operation are defined and a classification of operations is introduced.

Definition 1. Discrete manufacturing product dp

A discrete manufacturing product dp is defined by its set of attributes:

$$dp = \{\mathbf{att}_1, \ldots, \mathbf{att}_n\} \tag{1}$$

$$\mathbf{att} = (\text{label}, \mathit{value}_1, \ldots, \mathit{value}_s) \tag{2}$$

The label denotes the type of attribute. The number of values depends on the type of attribute. Attributes can be system-independent, attributes which describe product characteristics, as well as system-dependent, e.g. positioning attributes within the manufacturing system.

For example, if a bottle b is filled with corn, the bottle has the one-digit attribute $(\text{filled}, corn)$. The zero-digit attribute (empty) describes that the bottle is empty.

For modeling specifications of product attributes i.a. the ISO/IEC-compliant industry standard *eCl@ss* has been established [25].

In production technology the term *production process* (based on [26]) denotes a sequence of operations to convert a part from an initial to a final form based on engineering design information. The sequence of operations contains process description, process parameters, equipment and machine tools required for production.

The term *operation* in production technology is not clearly defined. In order to avoid ambiguities here the terms *production process* and *operation* are formalised as follows:

Definition 2. Production process p

Given is a set of input materials $In = \{dp_1, \ldots, dp_n\}$ and a modular CPPS consisting of a set of modules $M = \{m_1, \ldots, m_s\}$. Each module $m \in M$ is able to execute a set of potential operations $O_m = \{o_1, \ldots, o_d\}$. The set of all operation executable by M is $O = O_{m_1} \cup \cdots \cup O_{m_s}$.

Let dp_{end} be a final discrete manufacturing product. A production process p is a partially ordered set with a greatest element or totally ordered set (chain) of operations of O, so that In is converted into dp_{end} in *processing time* t_p.

The set of input materials contains all materials, which are needed during the product process to produce the final manufacturing product. The set of modules are technical resources of the manufacturing system. Technical resources are production modules but also transport modules. They execute operations on input materials and intermediate products:

Definition 3. Operation o

Given is a set of input products In and a module $m \in M$ of a set of modules of a CPPS. Let Out be a set of output products.

A process is called operation o of module m, if m processes In over a specific period of time, *processing time* t, into Out.

The set of input products of an operation includes input materials but also intermediate products. The output are intermediate products or end products. For example, the production module *put_module* has a stock with lids (input material). During the operation *putting a lid on a bottle* a lid is put on a corn-filled bottle (intermediate product).

Regarding the criteria change of time, positioning and product attributes different operations can be classified:

TABLE I: Classification of operations in a production process

Change of	Time	Positioning attributes	Product attributes
Storage operation	✓	-	-
Transport operation	✓	✓	-
Production operation	✓	-	✓
Production and transport operation	✓	✓	✓

Definition 4. Production operation o_{pr}

An operation is called production operation o_{pr} of a production module $m_{pr} \in M$, if m_{pr} machines In over processing time t into Out and one of the following cases applies:

$|In| = |Out| = 1$ One input product is converted to exactly one output product.

$|In| = 1$ **and** $|Out| > 1$ One input product is converted to more than one output product.

$|In| > 1$ **and** $|Out| = 1$ A set of more than one input product is converted to exactly one output product.

$|In| > 1$ **and** $|Out| > 1$ A set of more than one input product is converted to more than one output product.

For example, during the production operation *screwing a lid l on a bottle b* two input products are converted to *a bottle with lid*: $In = \{l, b\}$, $Out = \{b = \{(\text{has\_screwed}, l)\}\}$.

The definition of production operation is oriented according to the VDI/VDE guideline 3682 [4].

Note that in production technology process planning is understood as a task of determining the sequence of *production* *operations* without consideration of *transport operations* or *storage operations*. But with regard to automated planning and scheduling of production processes this consideration is essential.

Definition 5. Transport operation o_{tr}
Let $P = \{p_1, \ldots, p_s\}$ be the set of positions to reach by a transport module $m_{tr} \in M$. A bijective mapping f defines the movement of m_{tr} depending on processing time t.

$$f : P \times \mathbb{Q}^+ \to P, (p, t) \mapsto p' \tag{3}$$

$f(p, t_{start})$ specifies the mapping of start positions at the beginning o_{tr}. $f(p, t_{end})$ specifies the mapping of end positions at the end of o_{tr}.
An operation is called transport operation o_{pr} of m_{tr}, if m_{tr} moves In over processing time t. In and Out differs only in the positioning attributes of their products.

For example, during the transport operation *rotate* a bottle b changes its positioning attribute from $In = \{b = \{(\text{pos}, start)\}\}$ to $Out = \{b = \{(\text{pos}, screw\_module)\}\}$.

Definition 6. Production and transport operation $o_{pr\&tr}$
An operation is called production and transport operation $o_{pr\&tr}$ of a production and transport module $m_{pr\&tr} \in M$, if $m_{pr\&tr}$ machines and moves In over processing time t into Out.

Definition 7. Storage operation o_{sto}
An operation is called storage operation o_{sto} of a module $m \in M$ if m_{pr} neither machined nor moved In over processing time t: $In = Out$.

The modeling of production, transport, production and transport and storage operations becomes a crucial challenge for open AI planning to discrete manufacturing and vice versa to benchmarking manufacturing planning problems to AI planning.

IV. A Modeling Approach of Manufacturing Planning Problems

Modeling implies always to strike a balance between information, which has to be modeled and information which can be modeled. In relation to automated process planning and scheduling this means to strike a balance between an expressive modeling and an efficient automated planning. The selection of the planning system plays a key role, because the planning system determines, how the planning problem needs to be modeled.
In AI planning domain independent planners as well as domain dependent planners exist. Domain dependent planners are developed for exactly one planning domain e.g. one manufacturing system. They are unsuitable to cover the broad range of manufacturing systems in discrete manufacturing. In contrast, domain independent planners do not determine the planning domain and can be used to solve different manufacturing planning problems. Hence, domain independent planners are suggested for automated process planning and scheduling in discrete manufacturing.
Many existing domain independent planners support the standardised AI planning formalism PDDL. The modeling of planning problems in PDDL is subdivided into a domain, the description of possible actions, and a related problem, the description of the initial state of the planning domain and the desired goal (state of the domain). The domain file contains inter alia the description of *types, constants, predicates, functions* and *actions*. The problem file contains inter alia the description of *objects, init* and *goal*.

A. Domain and problem modeling

A manufacturing planning problem requires the modeling of the manufacturing system and the ordered products. The challenge is to determine, which aspects of the manufacturing planning problem have to be modeled in the domain and which aspects are part of the problem.

The domain of a manufacturing planning problem describes the manufacturing system: All possible operations executable by the different modules of the system. The problem of a manufacturing planning problem describes all information of day-to-day production: The current state of the system and the ordered products.
Particularly in relation to the domain a modeling decision has to be made. System structures can be modeled either in a general or specific way:

Modeling challenge 1. On the one hand operations can be described in a general way, such that the modeling of module skills is applicable independently to the specific manufacturing system. On the other hand the operations can be modeled specifically with respect to one manufacturing system design. Depending on the selected approach information need about the system structure needs to be included in the domain or the problem:
In the general approach information about the specific system structure is part of the problem description. Operations are described generally by means of parameters. E.g. a transport operation of a transport module determines the possible positions to reach, but does not define, which module is connected to which position. This approach has the disadvantage that with increasing number of variables the search space grows and automated planners fail to solve the planning problem. In a specific modeling approach the operations in the domain entail specific knowledge about the manufacturing system. Constants rather than parameters are used to model the operations. If the technical system is frequently modified, a specific modeling requires a certain modeling effort in case of changes. But beyond that a specific modeling approach increases the efficient generation of plans/schedules by automated planner. Th challenge ist to figure out, when a general modeling and when a more specific modeling is appropriate.

Modeling proposal 1. In general, all information related to the technical system are modeled in the domain file, so that changes of the technical systems imply the adaption of the domain modeling. All information related to day-to-day

production, e.g. available input materials, the current state of the system and ordered products, are modeled in the problem file, so that changes of e.g. orders or available input materials imply the adaption of the problem modeling.
Depending on the complexity of the manufacturing planning problem a specific or general modeling approach is selected. Manufacturing planning problems are usually complex planning problems, so that an efficient generation of plans/schedules by automated planners requires a specific modeling.

Example 1. In this section all modeling proposals are illustrated on a simplified example of the demonstrator *Versatile Production System (VPS)* of the SmartFactoryOWL: A rotary plate with four positions *p1 p2 p3 p4* connects two production modules. On position *p1* a bottle is put in. On position *p2* a lid is put on the bottle. On position *p3* the lid is screwed on. On *p4* the bottle is the output. The product operations *lid_putting* and *lid_screwing* and the transport operation *rotate* are executable by the VPS.
PDDL version 2.1 is used to model the VPS planning problem. The requirement *typing* is used to declare the type of *constants, objects* and *parameters*. Note, that the PDDL formalisme of attributes differs from the formalisation of attributes introduced in section III.

```
(define (problem VPS)
(:domain VPS)
(:objects b1 b2 b3 - bottle lid - closure)
(:init (applicable rotary plate) . . . )
(:goal (has_screwed b1 lid) . . . )
(:metric minimize (total-time)) )
```

```
(define (domain VPS)
(:requirements :durative-actions :typing :fluents . . . )
(:types bottle closure rot_position
        production_module transport_module)
(:constants r1 r2 r3 r4 - rot_position
            screw_module . . . - production_module
            rotary_plate - transport_module)
(:predicates (has ?b - bottle ?lid - closure) . . . )
(:functions (lid_stock ?lid - closure) - number)
(:durative-action rotate_a . . . )
(:durative-action lid_putting . . . )
(:durative-action lid_screwing . . . ) )
```

Example 1: PDDL formalisation of the VPS planning problem

B. Problem modeling

According to modeling proposal 1 the input materials, the ordered products and the current state of the manufacturing system are modeled in the problem file as presented here:

Input materials

Modeling challenge 2. Discrete manufacturing products are defined by their product attributes (see definition 1). During a production process input materials are processed to ordered products. Hence, the way of modeling input materials is essential. Two alternatives of modeling input materials exist. First, input materials are modeled individually as objects. They are individually labeled and defined by their individual product attributes. Second, input materials are modeled as quantity of material. Then they can simply modeled as function, but do not have an individual labeling. The challenge is to figure out, for which input materials an individual labeling is essential.

Modeling proposal 2. Input materials are modeled individually as objects, if they not serve to supplement a product. The modeling of the quantity of material suffices, if a individual labeling of items is not essential.

Example 2. The bottles *b1 b2 b3* are modeled as individual objects to identify them during the whole production process, whereas the function *lid_stock* defines how many lids are currently in stock.

```
(:objects b1 b2 b3 - bottle lid - closure)
(:init (= (lid_stock lid) 10) . . . )
```

Example 2: PDDL formalisation of input materials

Ordered products

Modeling challenge 3. Ordered products are defined by their product attributes. The challenge is in particular to model the significant product attributes, the critical knowledge about the ordered product. The modeling of ordered products depends i.a. on how input materials are modeled.

Modeling proposal 3. Ordered products are defined by the input materials and their product attributes, which they are supposed to fulfill at the end of the production process. Depending on how the input materials are modeled the product attributes of ordered products need to be defined.

Example 3. Three bottles are ordered: Two bottles *b1* and *b2* with a screwed on lid and one bottle *b3*, where the lid is put, but still not screwed. In this simple example each product is described by exactly one attribute.
Necessary precondition for *screwing a lid on a bottle* is *putting a lid on a bottle*. If a product has the attribute *has_screwed* this implies that also the product attribute *has* holds. In relation to the domain modeling VPS, it is not necessary to list the attributes *(has b1 lid)* and *(has b2 lid)*.

```
(:goal (has_screwed b1 lid) (has_screwed b2 lid)(has b3 lid))
```

Example 3: PDDL formalisation of ordered products

Current state of the manufacturing system

Modeling challenge 4. Modeling of a system comprises always to make a decision, which information is given explicitly and which information is implicitly presupposed. In any case, the modeling of the current state of the manufacturing has to include information about all products (input materials, intermediate and end products) and their current positioning and product attributes as well as the current applicability of modules. In case of module failure, than only this information needs to be adapted.
The more further explicit knowledge, e.g. the spatial arrangement of modules, is given, the more the complexity of the planning problem increases. The more implicit knowledge is assumed, the more expert knowledge is required in case of system modifications.

Modeling proposal 4. The description of all products and the applicability of modules represent the current state of the system. To restrict the search space of manufacturing planning problems the proposal is to model the spatial arrangement of modules implicitly by the operations in the domain.

Example 4. The transport module *rotary_plate* as well as the production modules *put_module* and *screw_module* are applicable. The rotary plate has four positions *r1* - *r4*.10 lids are in stock. On position *r1* bottle *b1* is located etc..
The knowledge on which position of the rotary plate which production module is located is implicitly given in the domain.

```
(:init (applicable rotary_plate)(applicable put_module)
       (applicable screw_module) (= (lid_stock lid) 10)
       (rotate_order_ position r1 r2 r3 r4)
       (on_position b1 r1)(on_position b2 r2)(on_position b3 r3))
```

Example 4: PDDL formalisation of the current system state

C. Domain modeling

Similar modeling challenges with regard to domain modeling exist. Solution approaches of modeling transport, production and storage operations are presented here.[1]

Transport operation

Modeling challenge 5. Transport operations vary in relation to how many products they transport at the same time and in what a way they transport the products. Depending on the number of products, which the transport system can move at same time, the complexity of modeling increases significantly. The challenge is to model this change of positions of products in a way that automated planner can deal with.
Take a transport module, e.g. a rotary plate, which changes 20 positions at the same time. All dependencies and change of positions can be explicitly modeled in relation to a reference base. Because of the size of search space it cannot be ensured that an automated planner can handle the modeling. In contrast an implicit modeling, where e.g. a reference base is not explicitly modeled, decreases the search scope of the planner.

Modeling proposal 5. A transport operation requires the modeling of positions changes. With increasing number of changing positions at a the same time, an implicit modeling of information is preferable to reduce the complexity of the operation for the automated planner.

Example 5. The *durative-action rotate* models the operation *rotate* by the transport module *rotary_plate*: The movement of four rotating positions *r1 r2 r3 r4* in relation to a fixed reference positions *p1 p2 p3 p4*. The predicate *rotate_order_position* implicitly specify that the first digit corresponds to reference position *p1*, the second to corresponds to reference position *p2* etc.. Depending on the current state of the rotary positions *r1 r2 r3 r4* at the beginning of the operation, in one time unit the rotary positions rotate one position further.

[1]The modeling production and transport operations will be introduced at a later stage.

```
(:durative-action rotate
 :parameters (?r1 ?r2 ?r3 ?r4 - rot_position)
 :duration (= ?duration 1)
 :condition (and (at start (and (not (activate rotary_plate))
                 (rotate_order_position ?r1 ?r2 ?r3 ?r4)))
                 (over all (applicable rotary_plate)))
 :effect (and (at start (activate rotary_plate))
         (at end (and (not (activate rotary_plate))
         (not(rotate_order_position ?r1 ?r2 ?r3 ?r4))
         (rotate_order_position ?r4 ?r1 ?r2 ?r3)))) )
```

Example 5: PDDL formalisation of a transport operation

Production operation

Modeling challenge 6. The execution of a production operation assumes that given input products have certain product attributes. E.g. a lid can be screwed on a bottle, if the bottle is located at the production module. Additional prerequisite for executing the operation is that a lid is put on the bottle. Such dependencies need to be modeled to guarantee the generation of correct production processes. The challenge is to figure out, which dependencies need to be modeled to avoid the generation of wrong production processes.

Modeling proposal 6. A production operation requires the modeling of essential attributes of input products i.a. the spatial positions of input products in relation to the executing production module and if necessary further product attributes.

Example 6. The production operation *screw* is executable, if a bottle is on position *p*3 (third position in the predicate *rotate_order_position*) and a lid is put on the bottle, but not screwed on: *(has ?b cup)(not(has_screwed ?b cup))*. At the end of the operation the processed bottle fulfills not only the attribute *has*, but also the attribute *has-screwed*.

```
(:durative-action screw
 :parameters (?r1 ?r2 ?r3 ?r4 - rot_position ?b - bottle)
 :duration (= ?duration 5)
 :condition (and (at start (and (not (activate screw_module))
                 (rotate_order_position ?r1 ?r2 ?r3 ?r4)(on_position ?b ?r3)
                 (has ?b cup)(not(has_screwed ?b cup))))
                 (over all (and (applicable screw_module)
                 (not (activate rotary_plate)))))
 :effect (and (at start (activate screw_module))
         (at end (and (not (activate screw_module))
         (has_screwed ?b cup)))) )
```

Example 6: PDDL formalisation of a production operation

Storage operation

Modeling challenge 7. Is a product neither machined nor moved storage operations apply. With regard to automated process planning and scheduling storage operations are important, because they affect the overall process planning and scheduling of ordered products on manufacturing systems. The challenge is to figure out, if an explicit modeling of storage operations is necessary.

Modeling proposal 7. Automated planners integrate automatically storage processes in the plans/schedules. An explicit modeling of storage operations is not required.

In this section different modeling approaches of manufacturing planning problems are discussed and a general approach of modeling input material, ordered products, system state, production, transport and storage operation is presented.

V. Evaluation

Different modeling approaches (MAs) discussed in section IV are evaluated by a case study. Benchmark is the VPS planning problem of the SmartFactoryOWL, because it represents a typical planning problem in discrete manufacturing. The planning domain consists of a rotary plate with 12 positions, a water filling module, a corn filling module, a cap putting and a cap screwing module, all connected to the rotary plate.
The planning system **Temporal Fast Downward (TFD)** [27], participating planner on the IPC 2014, is used on a macBookPro 2015, 2,2GHz i7, 16 GB to solve the VPS planning problem. The optimisation criterion of TDF is to minimise the makespan (maximum completion time). Three approaches of modeling the planning domain were evaluated. The MAs differ in the modeling of the rotary plate:

Modeling approach 1. The modeling proposals in section IV) are implemented in MA 1. The reference base of the rotary plate is modeled implicitly (see example 5 in IV). The placement of modules to the rotary plate is specifically defined by an implicit modeling (see example 6 in IV).

Modeling approach 2. A reference base of the rotary plate is modeled explicitly. The placement of modules is specifically defined.

```
(:durative-action rotate
 :parameters (?r1 ?r2 ?r3 ?r4 - rot_position) . . .
 :condition (and (at start (and (rotate_order_position ?r1 ?r2 ?r3 ?r4)
           (on ?r1 p1)(on ?r2 p2)(on ?r3 p3)(on ?r4 p4) . . . )) . . . )
:effect (and . . . (at end (and (rotate_order_position ?r4 ?r1 ?r2 ?r3)
        (on ?r4 p1)(on ?r1 p2)(on ?r2 p3)(on ?r3 p4) . . . ))) )

(:durative-action screw
 :parameters ( ?rp - rot_position ?b - bottle) . . .
 :condition (and (at start (and (module_position screw_module p2)
           (on ?rp p2) (on_position ?b ?rp) . . . )) . . . ) . . . )
```

Example 7: PDDL formalisation of MA 3

Modeling approach 3. The placement of modules is also modeled explicitly, but in contrast to MA2 the positions of the production module are generically defined in the domain. Specifically, the current positions of the modules are determined in the problem description.

```
(:durative-action screw
 :parameters ( ?rp - rot_position ?p - position ?b - bottle) . . .
 :condition (and (at start (and (module_position screw_module ?p)
           (on ?rp ?p) (on_position ?b ?rp) . . . )) . . . ) . . . )
```

```
(:init (module_position screw_module p2) . . . )
```

Example 8: PDDL formalisation of MA 3

MA1, MA2 and MA3 are evaluated on problems with a different number of orders (different planning problems). Orders are either a bottle filled with water or a bottle filled with corn.
In table II the first column represents the number of ordered products. TDF generates plans to a given problem so long as the optimal has been found. Each new generated plan has a better makespan as the previous one. Table II shows for each MA, the number of plans generated until an optimal solution has been found or 30 minutes have passed. In case of optimal solutions the total search time is presented.

TABLE II: Comparison of 3 MAs in relation to 5 problems

	MA 1		**MA 2**		**MA 3**	
# orders	# plans	sec	# plans	sec	# plans	sec
1	1	0.01	1	0.02	1	0.03
2	4	1.26	4	3.75	4	3.85
3	9	31.57	6	85.39	6	86.48
4	6	1800.00	7	1800.00	7	1800.00
5	10	1800.00	12	1800.00	12	1800.00

TFD solves for all MAs the given problems and generates plans/schedules. MA2 and MA3 do not differ except for the search times. For 1 and 2 orders the generated plans for MA1/MA2/MA3 not differ. For 3 and more orders the generated plans for MA1 and MA2/MA3 partially differ in sequence, in which the orders are processed. For 3 orders two different optimal plans for MA1 and MA2/MA3 are generated. For 4 orders the last generated plans, plan 6 of MB1 and plan 7 of MB2/MB3, are different but they have the same makespan. Plan 6 is generated after 120 seconds, plan 7 after 350 seconds. For 5 orders the first 9 plans are equal. The last generated plans, Plan 10 of MA1 and plan 12 of MA2/MA3 are different, but with the same makespan. Plan 10 is found after 1237 seconds, plan 12 after 1751 seconds. For 4 and more orders for all MAs TDF generates and optimises plans, but fails to prove an optimal plan within 30 minutes. Overall for all MAs the number of generated plans increases with rising number of orders.
To sum up, TDF generates more efficient plans for MA1 always faster than for MA2/MA3. There is never a better plan generated for MA2/MA3 than for MA1. The last generated plans for all problems and MAs have always the same makespan, but with increasing number of orders the performance for MA1 is much better.

TABLE III: MA1 - Detailed plan evaluation for 5 & 10 orders

	5 orders		**10 orders**	
	makespan	search t. in sec	makespan	search t. in sec
Plan 1	418	<10	645	10
Plan 2	403	<10		
Plan 3	388	<10		
Plan 4	358	<10	———	
Plan 5	344	<10		
Plan 6	333	<10	No further plans	
Plan 7	319	<10	within 30 min.	
Plan 8	313	<10		
Plan 9	310	70		
Plan 10	303	1237		

For the best performing modeling approach, MA1, table III shows the detailed evaluation for 5 and 10 orders. 10 orders correspond exactly to two times 5 orders. For every generated plan the makespan and the search time (from time t=0 sec, until when TDF found the plan) is presented.
According to table III the execution of two times Plan 10, thus two times 5 orders, takes 606 seconds. Hence, in any case an

optimal solution for 10 orders has not been found within 30 minutes. With the growing number of orders the complexity of the planning problem increases and the efficient generation of optimal plans becomes more difficult.

The evaluation shows: Depending on the modeling of the planning domain, the complexity of the planning problem increases and influences the efficiency of the planner. An implicit modeling approach, which means less restrictions to consider, improves the performance of the planner.

VI. Summary and further work

Contribution 1. A classification of operations in discrete manufacturing systems with regard to planning. The production operation, transport operation, production and transport operation and storage operation are introduced. It is determined, how products change during an operation.

Contribution 2. Based on the classification, a domain modeling approach for discrete manufacturing is presented. This modeling approach allows to model the manufacturing system and the ordered products as manufacturing planning problem in way so that automated planners efficiently generate process plans/schedules.

Contribution 3. The evaluation of tree different modeling approaches of the VPS planning problem of the SmartFactoryOWL are evaluated on the automated planning system TDF. It is shown that the proposed modeling approach has the best performance.

The paper builds a bridge between *automation* and *AI planning* and enables a realistic automated planning and scheduling in discrete manufacturing.

Further work is to evaluate additional manufacturing planning problems on different automated planners including a detailed empirical run-time analyse. Furthermore, a concept of replanning by automated planning process planning and scheduling is intended to improve the responsiveness to disturbances in dynamic manufacturing environments.

References

[1] H. Kagermann, W. Wahlster, and J. Helbig, *Deutschlands Zukunft als Produktionsstandort sichern: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0*, ser. Final Report of the Industrie 4.0 Working Group. Berlin: acatech – National Academy of Science and Engineering, 2013. [Online]. Available: http://forschungsunion.de/pdf/industrie_4_0_abschlussbericht.pdf

[2] NIST, "Strategic R&D opportunities for 21st century, Cyber-physical systems, Connecting computer and information systems with the physical world." National Institute of Standards and Technolog, USA, NIST report of the Steering Committee for Foundations and Innovation for Cyber-Physical Systems, 2013. [Online]. Available: http://www.nist.gov/el/upload/12-Cyber-Physical-Systems020113_final.pdf

[3] A. Bannat, T. Bautze, M. Beetz, J. Blume, K. Diepold, C. Ertelt, F. Geiger, T. Gmeiner, T. Gyger, A. Knoll, C. Lau, C. Lenz, M. Ostgathe, G. Reinhart, W. Roesel, T. Ruehr, A. Schuboe, K. Shea, I. S. genannt Wersborg, S. Stork, W. Tekouo, F. Wallhoff, M. Wiesbeck, and M. F. Zaeh, "Artificial Cognition in Production Systems," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 148–174, Jan 2011.

[4] VDI/VDE, "Formalisierte prozessbeschreibungen konzept und grafische darstellung," VDI/VDE-guideline 3682 part 1–2, May 2015.

[5] V. Vyatkin, "Software engineering in industrial automation: State-of-the-art review," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1234–1249, Aug 2013.

[6] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 979–991, 2009.

[7] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *Journal of Scheduling*, vol. 12, no. 4, p. 417, Oct 2008.

[8] VDI/VDE, "Fertigungsmanagementsysteme (Manufacturing Execution Systems – MES)," VDI/VDE-guideline 5600 part 1, october 2016.

[9] W. Shen, L. Wang, and Q. Hao, "Agent-based Distributed Manufacturing Process Planning and Scheduling: A State-of-the-art Survey," *Trans. Sys. Man Cyber Part C*, vol. 36, no. 4, pp. 563–577, 2006.

[10] N. Gupta and D. S. Nau, "On the complexity of blocks-world planning," *Artificial Intelligence*, vol. 56, no. 2, pp. 223 – 254, 1992.

[11] D. Mcdermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. D. Wilkins, "PDDL - The Planning Domain Definition Language," Yale Center for Computational Vision and Control, Tech. Rep., 1998. [Online]. Available: http://www.informatik.uni-ulm.de/ki/Edu/Vorlesungen/GdKI/WS0203/pddl.pdf

[12] M. Fox and D. Long, "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains," *Journal of Artificial Intelligence Research*, vol. 20, no. 1, pp. 61–124, dec 2003.

[13] S. Edelkamp and J. Hoffmann, "PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition," Tech. Rep., 2004. [Online]. Available: http://www.plg.inf.uc3m.es/ipc2011-deterministic/attachments/Resources/edelkamp-hoffmann-tr-2004.pdf

[14] A. Gerevini and D. Long, "Plan Constraints and Preferences in PDDL3 - The Language of the Fifth International Planning Competition," Technical Report, 2005. [Online]. Available: http://www.cs.yale.edu/homes/dvm/papers/pddl-ipc5.pdf

[15] D. L. Kovacs, "Complete bnf description of pddl 3.1 (completely corrected)," Tech. Rep., 2011. [Online]. Available: https://helios.hud.ac.uk/scommv/IPC-14/repository/kovacs-pddl-3.1-2011.pdf

[16] M. Fox and D. Long, "PDDL+ : Modelling Continuous Time-dependent Effects," 1999.

[17] M. Brenner, "A multiagent planning language," in *In Proc. of ICAPS'03 Workshop on PDDL*, 2003.

[18] D. Mcdermott, "The 1998 AI Planning Systems Competition," *AI Magazine*, vol. 21, pp. 35–55, 2000.

[19] M. Vallati, L. Chrpa, M. Grzes, T. L. McCluskey, M. Roberts, and S. Sanner, "The 2014 International Planning Competition: Progress and Trends," *AI Magazine*, 2015.

[20] V. Lifschitz, "On the semantics of strips," in *Reasoning About Actions & Plans: Proceedings of the 1986 Workshop*. Morgan Kaufmann, 1987, pp. 1–9.

[21] C. Fritz, "Automated Process Planning for CNC Machining," *AI Magazine*, vol. 37, no. 3, pp. 116–117, 2016.

[22] M. Helmert and H. Lasinger, "The scanalyzer domain: Greenhouse logistics as a planning problem," in *Proceedings of the 20th International Conference on Automated Planning and Scheduling, ICAPS 2010, Toronto, Ontario, Canada, May 12-16, 2010*, 2010, pp. 234–237.

[23] T. Niemueller, G. Lakemeyer, and A. Ferrein, "The RoboCup Logistics League as a Benchmark for Planning in Robotics," in *Proc. of Workshop on Planning and Robotics (PlanRob) at International Conference on Automated Planning and Scheduling (ICAPS)*, Jerusalem, Israel, 2015.

[24] P. Pittman, J. Blackstone, and J. Atwater, *APICS Dictionary, 15th Edition*. APICS, 2016.

[25] eCl@ss, "eCl@ss White Paper: Industry 4.0." URL: https://www.eclass.eu/fileadmin/downloads/ecl-Whitepaper-Industry40_EN_klein.pdf, last accessed 20 March 2018. [Online]. Available: https://www.eclass.eu/fileadmin/downloads/ecl-Whitepaper-Industry40_EN_klein.pdf

[26] B. W. Niebel, A. B. Draper, and R. A. Wysk, *Modern manufacturing process engineering*. McGraw-Hill, 1989.

[27] P. Eyerich, T. Keller, J. Aldinger, and C. Dornhege, "Preferring preferred operators in temporal fast downward," IPC 2014, 2014. [Online]. Available: https://helios.hud.ac.uk/scommv/IPC-14/repository/booklet2014.pdf