Capability-based Planning and Scheduling for Adaptable Manufacturing Systems

Nadine Keddis, Gerd Kainz, and Alois Zoitl
fortiss GmbH
Guerickestr. 25
80805 Munich, Germany
{keddis, kainz, zoitl}@fortiss.org

Abstract—**Manufacturing enterprises can only stay profitable if they manage to flexibly respond to changes in markets by adapting their products, product variants, and product volumes. To support such variety in products, we suggest a capability-based approach for production planning and scheduling. Production plans and machines are described in terms of required capabilities and provided capabilities respectively. Additionally, the topology of the factory is described. We propose to combine these descriptions to automatically generate production schedules that consider the material flow in the factory. For each production plan, our approach generates a valid schedule for the currently available machines in the factory without manually reconfiguring the software. The evaluation of the approach on an industrial production system used for educational purposes shows the suitability of this approach.**

I. Introduction

The manufacturing industry is currently under strong pressure to swiftly and easily adapt to changes. Mass customization is becoming the ideology of today's manufacturing [1], [2]. This is coupled with dynamic requirements regarding lot sizes, product variants, lead times, and costs which emphasize the need for adaptable production systems. Simultaneous production of different products is a main requirement for economic usage and a better utilization of production equipment. Increased adaptability while keeping setup times to a minimum is the key feature of future manufacturing [3]. By being adaptable, intelligent, and versatile, future manufacturing systems can deal with the turbulent environment that they face [4].

Today, production starts with a fixed production plan that is optimized for a specific product. The plant configuration is rarely changed once production is running. New products can only be scheduled after production has been stopped and some parts of it reprogrammed, reconfigured, or even replaced. Current production plans are unable to cope with the requirements of today's industry [5].

Scheduling in general is NP-hard and manufacturing scheduling is one of the most difficult to solve [6]. A lot of available scheduling algorithms are either very specific for a certain problem or too theoretical and therefore not applicable in real-world manufacturing systems [6]. Current planning and scheduling software requires a lot of manual effort [7]. For each available resource in the factory, jobs are assigned manually and the scheduling algorithm assumes that the factory topology is fixed. Moreover, most scheduling algorithms assume that the material will eventually arrive at the machine that should perform the operation. Therefore, they do not consider topology information during scheduling and do not calculate material flows. Furthermore, they often lack the flexibility to cope with future requirements. Approaches to automatically generate schedules are tailored for static environments and cannot handle dynamic changes in the scheduling problem [8].

In adaptable manufacturing systems, the setup of the factory describes which machines are currently used in the factory as well as how and where these machines are placed within the factory (e.g., which machines are connected to each other). Future production planning software should be able to cope with different factory setups. It should be possible to switch between different setups without manually reconfiguring the planning software every time a change occurs.

We suggest a capability-based planning approach to tackle these problems. In this approach, production plans are modeled independently of the current factory setup using capabilities. At the same time, factory setups, including material flow information and machine capabilities, are modeled as well. Capability-based in this context means that the necessary information for matching production steps and resources is described through capabilities of resources and required capabilities in each production step. Resources provide and implement capabilities that are required for the production. The required capabilities are described in production plans. The main contribution of this paper is the combination of these different models to calculate a production schedule that also considers the required material flow. The benefits of this approach are demonstrated on a modular production system. The approach allows the generation of valid production schedules for different factory setups without any manual effort in-between setup changes.

The paper is structured as follows: Section II evaluates existing approaches and gives an overview of related work in the field of planning for adaptable manufacturing systems. In Section III, we describe the modeling of production plans, production systems, and material flow information. The suggested planning and scheduling approach is described in Section IV. In Section V, the industrial setup is described and the approach is evaluated. Finally, Section VI summarizes the paper and outlines future work.

II. Related Work

In the past years, much work has been done in the field of scheduling in general and on adaptable manufacturing in particular. The focus there is on optimizing scheduling algorithms and using multiple objectives for scheduling. In the artificial intelligence domain, a lot of work has been done to improve planning. The goal in this case is to find a set of actions and a sequence of these actions that leads to the goal state [9]. However, the planned actions do not involve material and therefore the material flow is not considered.

One class of optimization algorithms for scheduling problems is based on genetic and evolutionary algorithms. Schedules are optimized according to multiple objectives (for example, see [10], [11]). While these algorithms are indeed useful for optimizing schedules, the scheduling problem has to be modeled manually by assigning each operation to a defined machine. The manual modeling requires time and effort. Also, the material flow is not modeled in the problem. These approaches are therefore less suitable for a frequently changing production environment.

There is plenty of work in the domain of agent-based planning and scheduling. Most of these approaches focus on defining a suitable platform for agents and the negotiation aspect of agent-based planning. They also consider planning strategies and reconfiguration (see, for example, [6], [12]–[20]). Holonic manufacturing approaches also follow this line of research (see, for example, [21]). In this work, the planning focus is on allocating resources for the different jobs in a distributed system. In agent-based planning, the material flow is always considered implicitly by using transport agents that negotiate possible material flows. However, these approaches negotiate the schedule stepwise and therefore do not determine the whole schedule until it is already scheduled on the machine. Hence, optimizing the generated schedule is not possible, because it is not known beforehand. Nevertheless, a lot of these approaches use an ontology to enable collaboration between agents. These ontologies can be used as well, since the definition of a vocabulary is one of the building blocks of our approach.

Capability-based approaches are becoming more popular, since they offer more flexibility. However, available approaches mostly have a different focus. *Ollinger et al.*, for example, use a service-oriented description to ease integration of components [22]. The advantage in this case is the ability to reuse control programs. *Järvenpää and Torvinen* also follow a capability-based approach, but use it to evaluate the impact of changes in adaptable manufacturing systems [23]. Here, the capability-based approach serves as an assessment function for the effort needed to adapt to a change.

Zäh et al. propose a similar approach [5] to ours. They also propose to describe production processes and resources in terms of capabilities. However, they do not model the material flow and the topology of the factory. Production plans are optimized locally on a machine level. Additionally, they suggest using Radio Frequency Identification (RFID) technology to store production plans and plan and schedule according to the information on the RFID transponders. However, RFID is not always suitable for real-world setups due to interference problems, read/write speed, and limited storage capacity [24].

In contrast to the presented approaches, our approach aims at automatically generating a production schedule that considers the material flow without having to manually model the problem whenever a change occurs. It does not locally optimize at the machine level but rather has a global view on the production process. It can later be combined with genetic and evolutionary approaches to further optimize production.

III. Capability-based Modeling of Production Systems

In order to increase the flexibility of the planning process, we propose to describe the scheduling problem in a machine-independent way. This can be achieved by first describing the required operations independently from the factory in a production plan. Afterwards, each machine in the factory has to be described using the same vocabulary that was used to describe the production plans. In addition, we suggest modeling the material flow. In this section, we give a brief introduction on how to describe production plans, machines, and material flow in the factory using a capability-based approach.

A. Defining the Description Vocabulary

In order to use such a capability-based approach, the production plans as well as the factory models have to be described using the same vocabulary. This is an essential prerequisite. The same vocabulary does not necessarily only mean the same words; the semantics behind the words have to be considered as well. Two descriptions only match when their semantics are the same. This can be achieved by using an ontology to describe the domain. There are some efforts to standardize descriptions of operations in the manufacturing domain. An example for this is the German guideline VDI 2860, “assembly and handling”. The definition of ontologies to describe all possible operations of a domain is outside the scope of this paper and requires further research. Instead, a simplified ontology is described here to illustrate the approach.

In our case, we define a set of primitive operations that are used in the discrete manufacturing domain and incorporate these in our description. Primitive operations are operations that describe a self-contained operation with all sensor readings and actuator control to fulfill an atomic function. Examples are drilling, transport, supply, testing, and assembling. Additionally, the primitive operations can be combined to describe more operations. These are called combined operations; a sorting operation is one example. Sorting could, for example, be the result of combining a conveyor belt with transport capabilities, a color sensor that detects the color, and a module that can change the direction of the products to sort them. However, the decision whether operations are primitive or combined depends on the definition and there is no standard yet. A description of an operation can then be further refined to include attributes like material, geometry of the material, and further parameters that specify the operation in detail. This is necessary because the same operation description might apply to different materials, but the operation cannot be done with the same tools for these materials (e.g., drilling plastic and drilling steel). In order to distinguish between different cases, the operations can be grouped in classes and later refined using this attribute mechanism. Table I gives an example of possible descriptions used in our setup.

TABLE I. PRIMITIVE AND COMBINED OPERATIONS USED FOR THE DEMONSTRATION SCENARIO

Capability	Description
Supply	Supply material such as cases, boxes, sensors, covers, etc. The material can be specified with attributes.
Test	A class of capabilities that has to be refined to describe what has to be tested (e.g., *Test Height* and *Test Orientation*).
Assemble	Putting two materials together (e.g., a pick & place operation).
Detect	A class of capabilities that has to be refined to describe what has to be detected (e.g., *Detect Workpiece* and *Detect Color*).
Store	Buffering a number of workpieces or products. Attributes can be used to describe the capacity and type of the buffer.
Sort	Sorting workpieces or products according to a sorting criterion. The sorting criteria can be specified using attributes.

B. Capability-based Modeling of Production Plans

The first step in our approach is to describe the production plan[1] for each product that has to be produced. The production plan decouples the production process from the factory setup. The description is based on the previously defined capabilities. In a first step, we determine all the capabilities that are required for production. The second step adds all dependencies between the required capabilities. The dependencies determine the order in which the operations matching the capabilities have to be scheduled to result in the correct product. An operation in this context refers to the implementation of the capability or the actual execution of the capability on a resource. Dependencies can be described by defining either the predecessor or the successor for each capability. Thus, each production plan is represented as a directed graph $P = (S, D)$ with a set of vertices S and a set of directed edges D. Each vertex $s \in S$ represents a required production step. The details of the production step are described through the properties of the vertex. A directed edge $d = (a, b)$ is an edge going from vertex a to vertex b. It represents the dependency between two production steps that are represented by their vertices. It means that production step a has to finish before production step b can start. Additionally, the edges indicate the required material flow.

For vertex s, the incoming edges are represented by the set $inEdges = \{d | d \in D \wedge \exists t \in S : d = (t, s)\}$. The outgoing edges of s are represented by $outEdges = \{d | d \in D \wedge \exists t \in S : d = (s, t)\}$, respectively. The $inEdges$ set represents the predecessor information of an operation. Each operation that is connected by an $inEdge$ to the operation represented by vertex s must be finished before s can start. Respectively, the $outEdges$ set represents the successor information of an operation. Operations connected by an $outEdge$ to vertex s can only start as soon as s finishes. It is sufficient to use only predecessor or successor information. Each vertex can have zero, one, or more incoming and outgoing edges ($|inEdges| \geq 0, |outEdges| \geq 0$) which represent the predecessors and successors respectively. The dependencies between steps are transitive, so they need to be specified only for the immediate predecessor or successor. The sources of P represent the initial steps in the production plan. Sources are vertices v_{so} with $|inEdges(v_{so})| = 0$. The last production step is the sink v_{si} of P with $|outEdges(v_{si})| = 0$. Currently, the approach is limited to acyclic production plans. In future work, adding cyclic dependencies should be possible. The structure of the production plan reveals information about the scheduling problem such as the degree of parallelism, the critical path of the production as well as the leeway in scheduling decisions for independent operations with no defined order. Three simplified examples are shown in Figure 1. Boxes represent the required capabilities and arrows the dependencies between them.

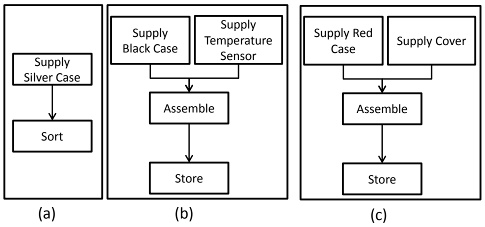

Fig. 1. Different production plans: (a) A silver workpiece, (b) a black thermometer, and (c) a red box.

C. Capability-based Modeling of Machines

Besides describing the production plans, the machines in the factory have to be described as well. The description should include all capabilities of a machine. Again, the description is based on the previously defined capabilities with their attributes, like material or duration. Machines are defined as a combination of different modules that implement basic independent operations that are described by capabilities. Each module implements one or more of the predefined capabilities. Additionally, combining modules can result in further capabilities as mentioned before. Figure 2 depicts an example for modeling a machine. For a detailed description of the factory model, see [25].

D. Material Flow Modeling

Since we are looking at adaptable manufacturing systems, modeling the material flow is a key step for automated planning. Most common planning approaches assume that the machines in a factory are fixed and therefore only map operations to machines. This also implies that the material will eventually arrive at the right time at the right station for the target operation without having to consider this during planning. For adaptable manufacturing systems, this assumption does not hold anymore because the setup of the system might change on a daily or even hourly basis. On the one hand, machines that were available at planning time might not be available anymore during production. On the other hand,

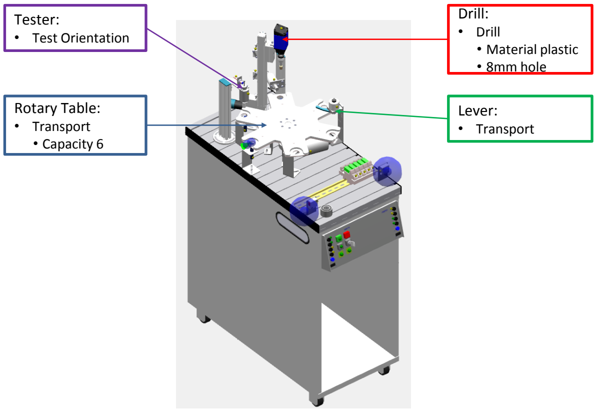

Fig. 2. A capability-based machine model with four modules providing the capabilities drill, test orientation, and transport.

[1] For simplification, this paper discusses only the specification of the production plan without errors. However, the suggested approach also allows the specification of actions that should be performed in case of an error.

new machines that were not considered in the initial planning might be available during production. In addition, the topology might change between different products. Multi-agent systems and holonic approaches tackle this challenge by negotiating scheduling decisions after each step [16]. This, however, leaves little room for optimizing schedules since the schedule is only known after actually performing all steps.

To overcome these problems, the material flow has to be modeled for every factory setup as well. This is achieved by describing input and output points of each machine in addition to its capabilities. Machines can also have connection points that are used as input as well as output points. We call these connection points bidirectional. The material flow is then modeled as connections between connection points of machines. The connections are directed connections, with their direction being determined by the connected points. The resulting material flow of a factory setup can be represented by a directed graph $M = (R, F)$ where R is the set of vertices representing available machines and F is the set of edges representing the material flow between the machines. A directed edge $f = (r_1, r_2)$ indicates that there is a valid material flow from machine r_1 to machine r_2. If we, for example, connect an input point of machine 1 with an output point of machine 2, then the direction of the connection is from machine 2 to machine 1. For bidirectional material flow, two edges $f_1 = (x, y)$ and $f_2 = (y, x)$ are used to represent that the material can flow in both directions. Vertices r with $|inEdge(r)| = 0$ either have no input or are not connected on their input. Vertices r with $|outEdge(r)| = 0$ either have no output or are not connected on their output. When connecting bidirectional points, the connection goes in both directions.

Figure 5 depicts examples of material flow models for three different factory setups. The first setup does not include bidirectional connections, whereas the other setups do. Since it is not feasible to model all possible material flows during layout planning, automated approaches are required. In earlier work, we suggested a plug&produce approach for this purpose (see [25]).

IV. Production Planning and Scheduling

The previously described capability models can be used as a starting point for automatic generation of production schedules. The production plan models and the capability-based descriptions of machines in the factory can be combined to determine a mapping from operations to machines. A simple matching algorithm can be used to determine such a mapping. The initial mapping can be refined to only consider possible material flows and hence only generate schedules that match the current factory setup. In this section, we introduce the algorithm that we developed based on the workflow depicted in Figure 3 that automatically generates production schedules based on the different capability-based models. It starts with a definition of the production plan for a specific product illustrated in the upper left corner. Based on the required capabilities and the factory setup (both shown in the lower left corner), a resource mapping can be calculated as seen in the upper right corner. This can be extended to only include solutions with valid material flow. The best schedule is then used to control the production.

A. Automated Generation of Production Schedules

Production planning consists of routing, scheduling, dispatching, inspection and coordination, control of materials, machines, tools, and operating times [26]. Initially, the production schedule has to address specific key elements in advance to ensure an uninterrupted flow of work. These elements are material ordering, equipment acquisition, bottlenecks, and human resource acquisition and training. The choice of the manufacturing process is affected by factors such as volume, variety, capacity of the plant, lead time, flexibility, and efficiency. We focus on the scheduling of the production since it plays an important role in the efficiency of the process and the cost and is necessary for allocating resources for the jobs [27]. Since the manufacturing constraints can change during the product life cycle, scheduling approaches have to be dynamic and capable of adapting to these changes in an efficient manner [28]. Hence, having an efficient way of scheduling will reduce the cost and make rescheduling less problematic and enable a more flexible and adaptable manufacturing system. This can be achieved through automatically generating production schedules that consider the current factory setup and the required efficiency criteria.

We suggest using a complete search with branch-and-bound and backtracking to generate schedules. Since the number of machines that have to be considered in each step is much smaller than the number of operations, the search space is small. Typically, machines are not used for all operations. Therefore, branches can be pruned quickly and the number of branches that reach the last level in the search tree is rarely exponential in practice. Thus, the search space is also not exponential. Using a complete search in this case results in the best solution. The solution space that is explored by the search is the automatically generated factory model as described in our previous work [25]. This is the previously described graph M. The factory model contains topology information of the current setup as well as a description of each machine in the factory with its capabilities as depicted in Figure 3 in the lower left box. The first step of the scheduling procedure results in a mapping of required capabilities to available machines. For each capability in the production plan graph P, a list of possible machines that provide the capability is generated. By iterating over all available machines in M and using a simple matching algorithm, such a list can be provided for each step in the production plan. As an example, the matching algorithm would assign machine “Pick&Place Temperature” and machine “Pick&Place Cover” from setup (a) in Figure 5 for capability “Assemble” in production plan (c) in Figure 1. The resource mapping is the starting point for the material flow calculation in the following step. Since the factory can have different topologies, the resource mapping is not sufficient. It is also necessary to check whether the material can flow between the chosen resources. This is achieved by generating schedules based on the resource mapping and the factory topology information. In order to maintain precedence relations between operations, the schedule is generated backwards, i.e., the last step of the production plan is inserted in the schedule first. However, it is inserted at the end of the schedule. The operations that are scheduled after are inserted at the beginning of the schedule. Thus, the starting point is the sink in graph P. This ensures that no operation is scheduled before its predecessors are finished.

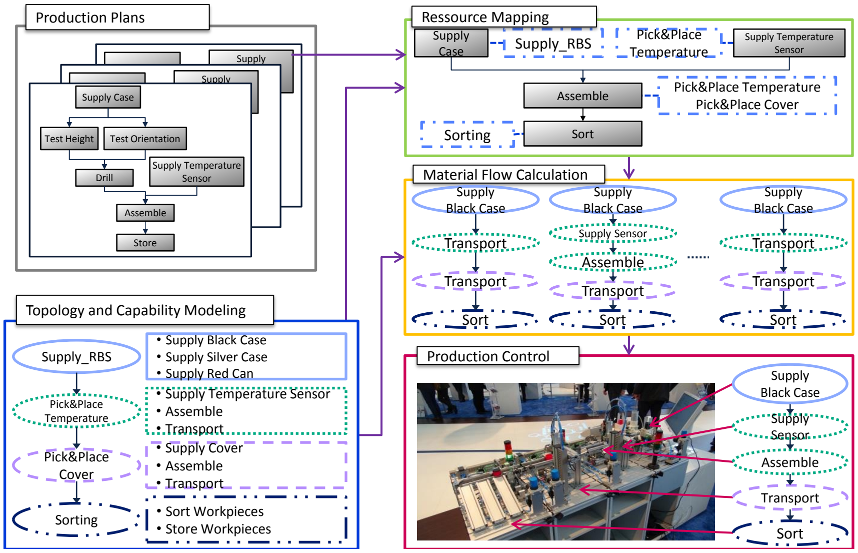

Fig. 3. Workflow for automatically generating production schedules.

Algorithm 1 describes the implementation of the approach. Operation *reverse graph* (line 4) reverses the direction of each edge in the graph. Operation *topological sort* (line 5) returns a linear ordering of the nodes such that for every edge (u, v), node u is before node v in the ordering. Operation $at(i)$ returns the data from the vector at index i. Variable $nextOp$ stores the reference to the following production step based on the production plan. Variable $stepNumber$ counts how many operations have already been scheduled to determine whether the search is finished or not. Variable c stores the current capability that should be scheduled next and variable m the machine that provides this capability. In line 22 operations $nextTask.machine$ and $data.machine$ return the machine that the next task and current task are scheduled on.

The search starts with the last operation o of the production plan (the sink of graph P). Operation o is scheduled on one of the machines that provide the capability. Each element in the branches set $B = \{m \in R | \exists c \in C : c = o\}$, where C is the set of capabilities of m and R the set of available resources, serves as a starting point for a branch of the search. Next, we try to schedule one of the predecessors of the scheduled operation. The available predecessors are represented by the set $N = \{s \in S | \exists d \in D : d = (s, o)\}$. We again choose one of the machines that provide the capability and try to schedule the operation on it. Besides providing the capability, the material must be able to flow from the current machine to the previously scheduled machine. If the operations are scheduled on the same machines, this is given. In all other cases we can check this using a breadth-first-search on the material flow model M (line 25). The search determines whether the material flow is possible or not and in the first case provides a list of all intermediate machines that have to transport the material so that it arrives at the right destination. For example if the "Pick&Place Cover" was chosen for the "Assemble" capability instead of the "Pick&Place Temperature", an additional transport operation has to be scheduled on the "Pick&Place Temperature" for the schedule to work correctly on setup (a) in Figure 5. The reason for this is that the previous capability "Supply Red Case" can only be assigned to machine "Supply_RBS" which is not directly connected to machine "Pick&Place Cover". The only way to transport the material from "Supply_RBS" to "Pick&Place Cover" is through "Pick&Place Temperature". Other graph search algorithms such as Dijkstra's algorithm are also possible here. Which algorithm to use depends on the criteria that are used to find a suitable path from one machine to the next. The steps to schedule an operation are repeated for all operations until a valid schedule is found. We can stop the search in case no machine exists that provides the capability or there is no material flow between the scheduled machines (line 28). In both cases we backtrack to the next unexplored branch (lines 14 - 16) and continue the search from there. When backtracking, the variable $stepNumber$ is reset to the backtracking point. If there are no backtracking points left, there is no valid schedule for the production plan with the current setup. In some cases, multiple valid production schedules might exist. In this case we choose the schedule with the shortest schedule duration. The duration can be calculated as the sum of the durations of each operation obtained from the machine models. Other efficiency criteria can be used to choose between different schedules. Possible criteria are, for example, makespan, utilization rate, energy efficiency, delivery time, or combinations of these. However, optimizing schedules is a very challenging task and is computationally hard. A lot of research is ongoing in this area and the topic is beyond the scope of this paper.

B. Production Control

The last step of our approach is automated control of the production. After a valid schedule is generated for each product that should be produced, the production can be started automatically. Each machine maintains a list of the operations that are scheduled on it. The lists are a result of splitting the generated schedule according to machines. When the system is ready, the central control system sends a "start production" signal to all machines. The machines start with the first operation in their list. Whenever a machine finishes an operation, it informs the supervisory control system which starts the next operation if there are operations left in the list. The production is finished when all operations are done.

```
Algorithm: Production Scheduling
input: Production plan as a precedence graph $P$
int $stepNumber = 0$;
Schedule $result$; Stack $st$;
$\overline{P}$ = reverse graph of $P$;
Vector $V$ = topological sort of $\overline{P}$;
Capability $c = V.at(0)$ ;
foreach machine $m$ with capability $c$ do
    $currentData.data$ = ($m$, $c$);
    $currentData.depth$ = 0; $st$.push($currentData$);
end
while !$st$.empty() do
    $data$ = $st$.pop();
    if backtracking necessary then
        backtrack and start next branch;
    end
    $currentStep$ = $V.at(stepNumber)$;
    $data$.nextOp = $P$.successor($currentStep$);
    if $stepNumber = 0$ then
        $result$.pushBack($data$);
    else
        $nextTask$ = find $data.nextOp$ in $result$;
        if $nextTask.machine$ = $data.machine$ then
            $result$.pushBack($data$);
        else
            if ∃ materialFlow from $nextTask.machine$
            to $data.machine$ then
                Schedule Transport operation for all
                intermediate machines;
                $result$.pushBack($data$);
            else
                No feasible solution for this branch,
                backtracking necessary;
            end
        end
    end
    if $stepNumber$ = $V$.size() then
        return $result$;
    else
        $o$ = $V.at(stepNumber + 1)$;
        foreach machine $m$ with capability $c$ do
            $currentData$.data = ($m$, $c$);
            $stepNumber$ = $stepNumber]1$ ;
            $st$.push($currentData$);
        end
    end
end
```

Algorithm 1: Production scheduling algorithm

V. Application Example

In this section, we describe the experimental setup that was used to evaluate the approach. Additionally, we present the results of the experiments that were conducted.

A. Experimental Setup

To illustrate the approach, we use a simplified production system designed for educational purposes. It is a Festo modular production system shown in Figure 4.

The setup consists of different stations, a conveyor belt, and a mobile robot that can be combined to form the desired manufacturing system. Stations can be arbitrarily combined. However, some combinations do not result in valid setups. Figure 5 illustrates three of the possible factory setups. The manufacturing system can produce temperature sensors in three different colors: black, red, and silver. It can also produce red boxes. Transporting raw material, like black, red, and silver cases and red cans, within the manufacturing system is also possible. Not all factory setups support all production plans. The different steps of the production processes include distributing of material, testing, processing, assembling, storing, and delivering.

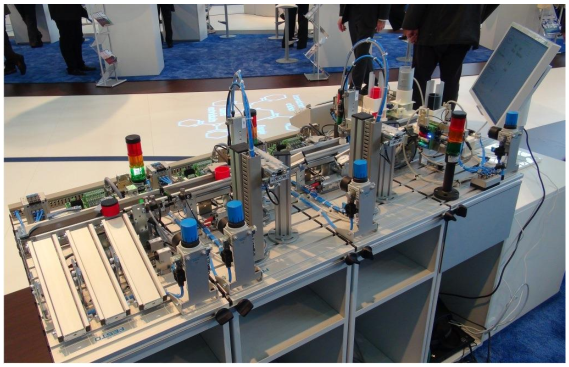

Fig. 4. One possible setup used for demonstrating the approach.

B. Results

For each of the setups we tested the approach on three different products. For setup (a) in Figure 5, a simple black thermometer, a red box, and a silver workpiece were produced. The production plan for each of the products is displayed in Figure 1. The same products were tested for setup (c). Since setup (b) offers the least product variability and is a production line similar to setup (a), the results are omitted here. Table II illustrates the duration in time units for each of the operations on the different machines. For the sake of simplicity, we assume that the robot always requires the same amount of time for transporting material from one machine to another.

After calculating a valid material flow for each of the products, the different products are scheduled using a least slack time strategy. The deadline for all products is assumed to be 100s. The execution time e of one schedule for a product can then be calculated as the sum of execution times of all operations. The slack time can be calculated according to the formula $(d-t)-e$, where d is the deadline, t the current time step, and e the execution time. The results obtained for setup

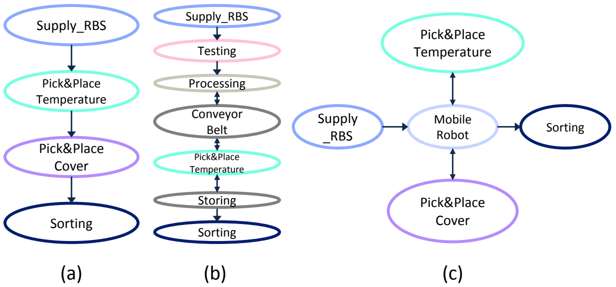

Fig. 5. (a) Setup with four different stations. (b) Setup with six different stations and a conveyor belt. (c) Setup with four different stations and a mobile robot.

TABLE II. THE DURATION OF OPERATIONS ON DIFFERENT MACHINES IN TIME UNITS.

Machine	Operation	Duration
Supply_RBS	Supply Case	9s
Pick&Place Temperature	Supply Sensor	1s
Pick&Place Cover	Supply Cover	1s
Pick&Place Temperature	Assemble	6s
Pick&Place Cover	Assemble	8s
Pick&Place Temperature	Transport	5s
Pick&Place Cover	Transport	5s
Sorting	Sort	5s
Robot	Transport	7s

(a) are illustrated in Figure 6. This also represents the optimal solution for this problem.

In setup (c), a robot is used for all transport operations. We again use a least slack time strategy, but without intertwining different products on a resource. Since the same algorithm as for assembly lines is used, all products must be scheduled sequentially and products cannot overtake each other during production. This means that each resource finishes all the steps for one product before it starts with the next product. The resulting schedule is shown in Figure 7 in the upper chart. It is not as efficient as the optimal one, which is shown in the lower part of Figure 7. Using cell production rather than an assembly line allows for mixing products and does not require sequential execution; the optimal solution takes advantage of this, while the generated schedule does not. This can be addressed by using a better scheduling strategy that fills the idle gaps for the robot and allows different products to be intertwined.

C. Evaluation

With this demonstration setup, we showed that we can automatically generate production schedules without reprogramming or manually reconfiguring the control software. The same algorithm can be applied for different factory setups and different production plans. This is achieved by considering the topology of the factory in the planning and calculating different possible material flows that result in a valid schedule. Without further interaction with operators, the schedule can be generated based on the production plans, the capability-based models of the machines as well as the material flow model. The generated schedule is optimal in case of an assembly line. For cell manufacturing with a robot for the transport operation, the generated schedule is valid but not optimal. Additional criteria need to be defined in order to make this schedule optimal. Using the capability-based approach to model production plans decouples the workflow from the factory setup and enables automated planning for different factory setups. For small-sized problems with six machines and production plans that include up to ten steps, the planning algorithm can generate schedules for different products within seconds and start the production. This is especially useful for adaptable factories where the setup is changed more frequently. This approach still has to be evaluated on larger problems to determine how it scales for industrial setups.

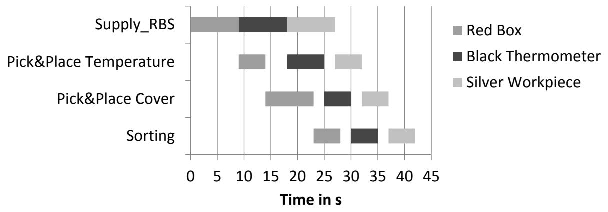

Fig. 6. Result of the scheduling using least slack time for an assembly line.

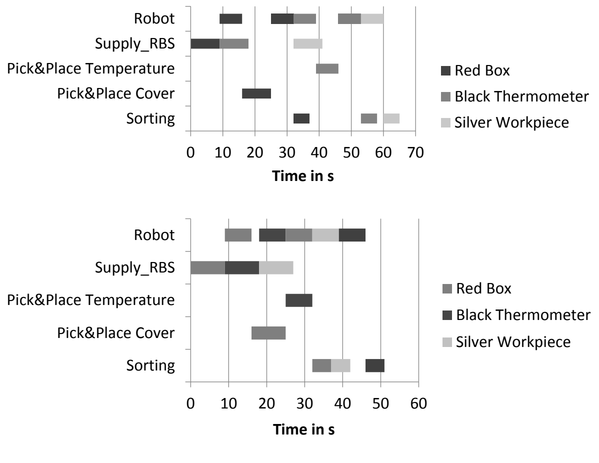

Fig. 7. Scheduling result using least slack time (top chart) and optimal schedule (bottom chart).

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a capability-based planning and scheduling approach for adaptable manufacturing systems. Besides describing the production process in terms of capabilities, this approach considers topology information during planning to calculate schedules with valid material flow. This enables automatic generation of schedules for different factory layouts without manual intervention, which is especially beneficial for adaptable manufacturing where the factory setup changes frequently. We used a simplified industrial setup to show how our approach can be applied. The setup demonstrates that we can generate schedules for different production plans and different factory layouts – and automatically execute the production process according to such a schedule – without any manual changes to the plant control software.

The next step in our work involves evaluating the approach on larger scheduling problems. Additionally, we intend to include the state of machines for scheduling decisions. This is especially interesting when there are shared resources and duplicate machines. We also plan to look at optimizing the order of production schedules according to user-defined criteria. The idea is to find a scheduling strategy that works well for both assembly lines and production cells with inter-logistic problems.

AKNOWLEDGMENT

This approach is a result of the research and development project "AutoPnP", which was funded by the German Federal Ministry for Economic Affairs and Energy as part of the technology programme "Autonomics - Autonomous, simulation-based systems for small and medium-sized enterprises" and supervised by the German Aerospace Center (DLR) under Grant No. 01MA11002.

REFERENCES

[1] Iacocca Institute, "21st Century Manufacturing Enterprise Strategy: An Industry-Led View," Iacocca Institute, Bethlehem, Pennsylvania, Tech. Rep., 1991.

[2] S. Hu, X. Zhu, H. Wang, and Y. Koren, "Product Variety and Manufacturing Complexity in Assembly Systems and Supply Chains," *CIRP Annals-Manufacturing Technology*, vol. 57, no. 1, pp. 45–48, 2008.

[3] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, "Reconfigurable manufacturing systems," *CIRP Annals-Manufacturing Technology*, vol. 48, no. 2, pp. 527–540, 1999.

[4] S. Zor, K. Görlach, and F. Leymann, "Using BPMN for Modeling Manufacturing Processes," in *Proceedings of the 43rd CIRP International Conference on Manufacturing Systems*, 2010, pp. 515–522.

[5] M.-F. Zäh, M. Beetz, K. Shea, G. Reinhart, O. Stursberg, M. Ostgathe, C. Lau, C. Ertelt, D. Pangercic, T. Rühr *et al.*, "An Integrated Approach to Realize the Cognitive Machine Shop," in *Proceedings of the 1st International Workshop on Cognition for Technical Systems*, 2008, pp. 6–8.

[6] W. Shen, L. Wang, and Q. Hao, "Agent-based Distributed Manufacturing Process Planning and Scheduling: a state-of-the-art survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 36, no. 4, pp. 563–577, 2006.

[7] B. Klopper, S. Honiden, J.-P. Pater, and W. Dangelmaier, "Decision Making in Adaptive Manufacturing Systems: Multi-objective Scheduling and User Interface," in *2011 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, 2011, pp. 123–130.

[8] M. Cheeseman, P. Swann, G. Hesketh, and S. Barnes, "Adaptive Manufacturing Scheduling: A Flexible and Configurable Agent-based Prototype," *Production Planning & Control*, vol. 16, no. 5, pp. 479–487, 2005.

[9] B. Klöpper, C. Sondermann-Wölke, C. Romaus, and H. Vöcking, "Probabilistic planning integrated in a multi-level dependability concept for mechatronic systems," in *IEEE Symposium on Computational Intelligence in Control and Automation (CICA), 2009*. IEEE, 2009, pp. 104–111.

[10] J. Chen and S. Ho, "A Novel Approach to Production Planning of Flexible Manufacturing Systems Using an Efficient Multi-objective Genetic Algorithm," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 7-8, pp. 949–957, 2005.

[11] J. Vidal, M. Mucientes, A. Bugarin, and M. Lama, "An Adaptive Evolutionary Algorithm for Production Planning in Wood Furniture Industry," in *International Symposium on Evolving Fuzzy Systems, 2006*. IEEE, 2006, pp. 267–273.

[12] W. Lepuschitz, B. Groessing, M. Merdan, and G. Schitter, "Evaluation of a Multi-Agent Approach for a Real Transportation System," in *IEEE International Conference on Industrial Technology (ICIT), 2013*. IEEE, 2013, pp. 1273–1278.

[13] T. Gabel and M. Riedmiller, "Adaptive Reactive Job-Shop Scheduling with Reinforcement Learning Agents," *International Journal of Information Technology and Intelligent Computing*, vol. 24, no. 4, 2008.

[14] C. Alexakos, M. Georgoudakis, A. Kalogeras, and S. Likothanassis, "Adaptive Manufacturing Utilizing Ontology-driven Multi-Agent Systems: Extending Pabadis' Promise Approach," in *IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2012, pp. 42–47.

[15] Y. Alsafi and V. Vyatkin, "Ontology-based Reconfiguration Agent for Intelligent Mechatronic Systems in Flexible Manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 4, pp. 381–391, 2010.

[16] C. Wang, H. Ghenniwa, and W. Shen, "Real Time Distributed Shop Floor Scheduling Using an Agent-based Service-oriented Architecture," *International Journal of Production Research*, vol. 46, no. 9, pp. 2433–2452, 2008.

[17] X. F. Zha, S. Y. E. Lim, and W. F. Lu, "A Knowledge Intensive Multi-Agent System for Cooperative/Collaborative Assembly Modeling and Process Planning," *Transactions of the SDPS, Journal of Integrated Design and Process Science*, vol. 7, no. 1, pp. 99–122, 2003.

[18] A. Bratukhin and T. Sauter, "Distribution of MES Functionalities for Flexible Automation," in *8th IEEE International Workshop on Factory Communication Systems (WFCS), 2010*. IEEE, 2010, pp. 157–160.

[19] S. Bussmann and K. Schild, "An agent-based approach to the control of flexible production systems," in *Proceedings of the 8th IEEE International Conference on Emerging Technologies & Factory Automation, 2001.*, vol. 2. IEEE, 2001, pp. 481–488.

[20] 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.

[21] D. C. McFarlane and S. Bussmann, "Holonic Manufacturing Control: Rationales, Developments and Open Issues," in *Agent-based Manufacturing*. Springer, 2003, pp. 303–326.

[22] L. Ollinger, J. Schlick, and S. Hodek, "Leveraging the Agility of Manufacturing Chains by Combining Process-oriented Production Planning and Service-oriented Manufacturing Automation," in *Proceedings of the 18th IFAC World Congress*, 2011.

[23] E. Järvenpää and S. Torvinen, "Capability-based Approach for Evaluating the Impact of Product Requirement Changes on the Production System," in *Advances in Sustainable and Competitive Manufacturing Systems*. Springer, 2013, pp. 173–185.

[24] P. Vrba, F. Macurek, and V. Mařík, "Using Radio Frequency Identification in Agent-based Control Systems for Industrial Applications," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 331–342, 2008.

[25] N. Keddis, G. Kainz, C. Buckl, and A. Knoll, "Towards Adaptable Manufacturing Systems," in *IEEE International Conference on Industrial Technology (ICIT), 2013*. IEEE, 2013, pp. 1410–1415.

[26] L. Benyoucef, B. Grabot *et al.*, *Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management*. Springer, 2010.

[27] J. O'kane, "A knowledge-based system for reactive scheduling decision-making in FMS," *Journal of Intelligent Manufacturing*, vol. 11, no. 5, pp. 461–474, 2000.

[28] E. Carpanzano, A. Orlandini, A. Valente, A. Cesta, and R. Rasconi, "Closed-loop production and automation scheduling in RMSs," in *Proceedings of the 16th IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*, 2011, pp. 1–8.