http://dx.doi.org/10.3182/20110828-6-IT-1002.02725

Planning transport sequences for flexible manufacturing systems [⋆]

Patrik Magnusson [*] **Nina Sundström** [*] **Kristofer Bengtsson** [**]
Bengt Lennartson [*] **Petter Falkman** [*] **Martin Fabian** [*]

[*] *Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden (e-mail: patrik.magnusson@chalmers.se)*
[**] *Research and Development, Teamster AB, Göteborg, Sweden (e-mail: kristofer.bengtsson@teamster.se).*

Abstract: When designing a manufacturing system it is important to plan what the system should do. One important activity in most manufacturing systems is to transport products or resources between different positions. In a flexible manufacturing system it can be challenging to design and plan these transport operations due to their complex logical behavior. This paper presents a method that identifies, creates and visualizes these transport operations based on inputs from a standard virtual manufacturing tool and a high level product operation recipe. The planning of the created transport operations is transformed into a problem of finding a non-blocking solution for a discrete model of the product refinement.

Keywords: Sequence planning, Supervisory control, Manufacturing operations

1. INTRODUCTION

Flexible manufacturing systems (FMSs) were early recognized as an interesting research topic, e.g. (Stecke, 1985) and (Buzacott and Yao, 1986). Especially scheduling and planning problems (Pinedo, 2009) and deadlock avoidance (Li et al., 2008) have been studied. Even if flexible manufacturing have been studied thoroughly by academia, the industry still struggles when developing these systems (Kouvelis, 1992), (Chryssolouris, 2006). The industrial focus has instead been on representing, visualizing and simulating the systems in so called virtual manufacturing tools, like Process Simulate from Siemens and Delmia Automation from Dassault Systemes.

One important activity when designing an FMS in industry is to plan the operation sequences (Chryssolouris, 2006). Methods that help the engineers to better understand operation sequences would therefore be an important contribution to their daily work. Especially methods that automatically identify and visualize the consequences of product and manufacturing design decisions on the operations would be useful. It is also important that a method is fully integrated in currently used virtual manufacturing tools. A method will be presented in this paper that can be implemented as a plugin in currently available virtual manufacturing tools, which identifies, creates, and visualizes transport operation sequences.

A part of an FMS is the automatic material handling system that transport products in between the workstations. The control design of the material handling system, i.e. creating transport operations, tends to be quite complex in modern FMSs. When a transport operation can execute is not only related to the layout of the manufacturing system, but also to the product design and the global state of the system. The planning is further complicated due to changes in product and manufacturing system design throughout the development process.

The suggested solution to the transport control problem by academia is usually to create a formal model, for example by using automata or Petri nets (Adlemo et al., 1995), (Li et al., 2008), and then verify and synthesize a correct supervisor or controller to avoid deadlocks and to guarantee a correct execution. Methods that uses formal models have, however, been hard to introduce in industry. The challenge tackled by this paper is therefore how to create correct transport operations, where the engineers do not need to change their working method or to bother about formal modeling.

This paper introduces a method that enables the engineers to use their normal design process and tools, yet using the formal framework supervisory control theory introduced by Ramadge and Wonham (1989). Based on where the products can be positioned in the FMS defined in a virtual manufacturing tool and a high-level product recipe, formal models are generated and a supervisor is synthesized. This supervisor is then translated back to the engineering tools as transport operations, which include correct conditions for when the operations can execute. This automated method can iteratively refine the operation conditions throughout the development process, hence the consequences of (late) changes will be directly updated in the transport operations.

The preliminaries are introduced in the next Section. An overview of the proposed method and an example product to be refined in a manufacturing system are presented in Section 3. Section 4 describes how to create operations and their initial conditions. In Section 5 the transport planning is discussed and the paper is concluded in Section 6.

[⋆] This work has been carried out within the Sustainable Production Initiative, the Wingquist Laboratory VINN Excellence Centre, and the Production Area of Advance at Chalmers. It has been supported by the European 7th FP, grant agreement number 213734 (FLEXA), Vinnova, and the Knowledge Foundation. The support is gratefully acknowledged.

2. PRELIMINARIES

The graphical language Sequences of Operations (SOP) introduced in Lennartson et al. (2010) will be used to visualize relations among the operations in the sequence planning tool Sequence Planner (SP) (Bengtsson, 2009). This tool uses a new sequence planning approach, where sequences are based on the relations among operations instead of explicit manual sequence construction. This is achieved by using self-contained operation models that include only relevant conditions on when and how the operations can execute. By the use of different views or perspectives the sequences of operations related to e.g the part flow, transport operations or workstation tasks can be visualized. It is important to stress that, in this paper, the term sequence not necessarily refers to a straight chain without alternatives (Lennartson et al., 2010).

An operation may formally be modeled as an Extended Finite Automaton (EFA) (Sköldstam et al., 2007).

Definition 1. (Extended Finite Automaton). An extended finite automaton is a 7-tuple

$$E_k = \langle Q_k \times V, \Sigma_k, \mathcal{G}_k, \mathcal{A}_k, \rightarrow_k, (q_k^0, v^0), M_k \rangle$$

The set $Q_k \times V$ is the extended finite set of *states*, where Q_k is a finite set of *locations* and V is the finite domain of definition of variables, Σ_k is a nonempty finite set of *events* (the alphabet), $\mathcal{G}_k$ is a set of *guard* predicates over V, $\mathcal{A}_k$ is a collection of *action* functions from V to V, $\rightarrow_k \subseteq Q_k \times \Sigma_k \times \mathcal{G}_k \times \mathcal{A}_k \times Q_k$ is a transition relation, $(q_k^0, v^0) \in Q_k \times V$ is the initial state, and $M_k \subseteq Q_k$ is a set of *marked (desired) locations.* □

This paper follows the notation in Lennartson et al. (2010) grouping guards and actions into a set C of transition conditions. These transition conditions include both the current and the next values of the variables after a transition. The variable values after a transition (next values) are denoted $v' \in V$. The conditions are a mapping from $V \times V \rightarrow \mathbb{B}$, $v \in V, v' \in V$. Consider e.g. a guard $v_1 = 0$ combined with an action $v_1 := 1$, which is expressed as one transition condition $(v_1 = 0) \wedge (v_1' = 1)$.

The EFA model of an operation may now be defined.

Definition 2. (EFA model of an operation). An operation may be modeled formally by an EFA where the set of locations $Q_k = \{O_k^i, O_k^e, O_k^f\}$, the event set $\Sigma_k = \{O_k^{\uparrow}, O_k^{\downarrow}\}$, the set of transition conditions $C_k = \{C_k^{\uparrow}, C_k^{\downarrow}\}$, the transition relation $\rightarrow_k = \{\langle O_k^i, O_k^{\uparrow}/C_k^{\uparrow}, O_k^e\rangle, \ \langle O_k^e, O_k^{\downarrow}/C_k^{\downarrow}, O_k^f\rangle\}$, the initial location $q_k^0 = O_k^i$, and all locations are marked $M_k = Q_k$, see Figure 1. □

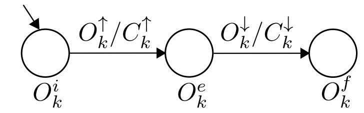
O_k
$O_k^{\uparrow}/C_k^{\uparrow}$
$O_k^{\downarrow}/C_k^{\downarrow}$
O_k^i
O_k^e
O_k^f

Fig. 1. Formal model of an operation O_k.

The basic assumption is that all operations, modeled for a system, are running in parallel. This is modeled by full synchronous composition, defined for EFA in Sköldstam et al. (2007). Relations among operations that restrict parallel execution are introduced in the transition condition for each operation. Any variable defined in V is implicitly included in all EFA models. The location set for each EFA model is implicitly included in V. Thus, predecessor requirements among operations are easily expressed. The term *SP model* refers to all EFA models of a system maintained in SP.

$C_k^{\uparrow}$ and $C_k^{\downarrow}$ are termed *pre-* and *post-condition* respectively. $C_k^{\uparrow} := X$ denotes that the pre-condition $C_k^{\uparrow}$ is set to X. $C_k^{\uparrow} := C_k^{\uparrow} \wedge X$ denotes that the updated pre-condition $C_k^{\uparrow}$ is a conjunction of the old pre-condition and X. An operation O_k is *satisfied* when its pre-condition $C_k^{\uparrow}$ evaluates to true, i.e. the system is in a state where the operation may be *executed.* Finally, an operation is *realized* in or by a resource.

3. THE PROPOSED METHOD

The proposed method is divided into three iterative steps: create operations, add local execution conditions and finally extend the execution conditions to achieve a correct global behavior. The operations that are created first are the processing operations, which specifies how the product is manufactured. These operations are created with the graphical SOP-language (Lennartson et al., 2010) in SP (Bengtsson, 2009). In parallel to this the layout of the flexible manufacturing system (FMS) is built up in a standard virtual manufacturing tool. The engineer will then assign what resources that will realize the operations and possible positions of the product directly in the virtual cell in the virtual manufacturing tool. To include demands through clicking on resources in a virtual cell is more appealing than drawing discrete models or write formal specifications. Transport operations are automatically generated and imported in SP, based on the layout of the FMS.

After the operations are created, (initial) execution conditions are added automatically to the operations based on which resources that are selected to realize the processing operations. An additional type of (planning) conditions are introduced automatically in order to achieve a correct global behavior. These (planning) conditions will, in many cases, introduce blocking situations, which are managed by the last step of the method. The last step uses the supervisory control theory (SCT) introduced by Ramadge and Wonham (1989) to remove blocking, deadlock, situations. The (planning) execution conditions assure that the supervisor calculation also results in a solution to the transport planning problem. The solution consist of transport operations with correct execution conditions, i.e. a satisfied operation can always execute and the system will never deadlock. These operations are visible in the graphical SOP-language in SP.

A product and a manufacturing system will be studied throughout this paper. This is both to demonstrate the complexity of planning transport operations and also to concretize the steps of the method.

The product to be refined comprises one part and one movable fixture. The part is illustrated in Figure 2. The product design requires three refinement processes of the part. These processes are modeled by the three operations O_A, O_B, and O_C. The part needs to be fixated to (unfixated from) the movable fixture before (after) the refinement operations. This is modeled by Operation O_F and O_U.

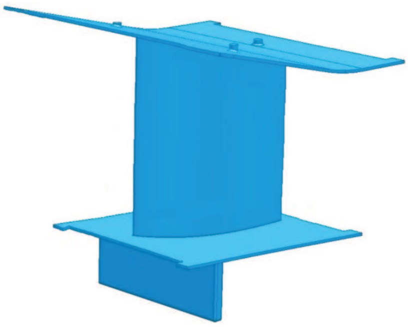

Fig. 2. The part to be refined in the FMS in the example.

The example FMS contains six *positions* where the product can be located in, five workstations and one buffer, which can be seen in Figure 3. Workstation 1 and 2 are in- and output to the cell while Workstation 3, 4, and 5 are processing machines. Furthermore, the cell contains a mover. The mover is a robot on a conveyor that may move and place products in the FMS. Operation O_F and O_U are realized either in Workstation 1 or 2. Operation O_A, O_B, and O_C are realized in Workstation 3, 4, and 5 respectively.

The sequential relations between the operations refining the product are visualized according to the SOP-language in Figure 4. O_F is the first operation to be executed. Operations O_A, O_B, and O_C may thereafter be executed in arbitrary order. O_U is the last operation to be executed, it may start when O_A, O_B, and O_C have been finished.

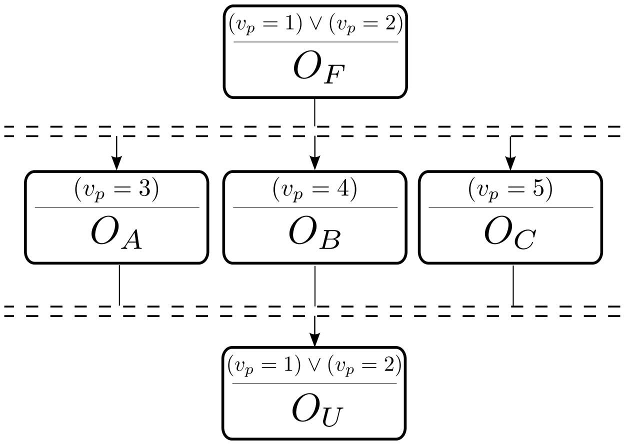

Fig. 4. View of process operations in example. Operation (O_U) O_F models (un-)fixation between part and movable fixture. (Un-)Fixation is realized in Workstation (1 or 2) 1 or 2. The operations O_A, O_B, and O_C model three refinement processes, to be executed in arbitrary order. These are realized in Workstation 3, 4, and 5 respectively.

4. REQUIREMENT SPECIFICATION IN A VIRTUAL MANUFACTURING TOOL

The five operations described so far are termed *process operations*. From now on, a product is described through its process operations. This paper proposes that mapping between process operations and workstations is performed within a virtual manufacturing tool like Process Simulate (PS) (Process Simulate, 2010), in parallel with the virtual cell modeling of the FMS. A plug-in with a GUI is programmed and implemented in PS (Sundström, 2010). To specify workstation(s) for realization of a process operation in PS the virtual workstation(s) is selected in the virtual cell, see Figure 3. This will add a pre-condition, to the operation, that requires the product to be in the selected workstation(s) before the operation may execute. The procedure is repeated for all process operations. The virtual resources are the positions for an FMS. A variable is introduced in the SP model to represent the positions. The current value of the variable points to the current position of the product. The selection of workstations is mapped to what value(s) the position variable must hold to satisfy the pre-condition of an operation in the SP model. The physical placing of the workstations is of no interest for a capacity mapping like this. The FMS layout may therefore be changed without affecting the mapping between process operations and virtual resources.

In the example, process operation O_A is realized in Workstation 3. To specify this information in PS process operation O_A is selected from a list of operations and then Workstation 3 is selected in the virtual cell. This results in a pre-condition for O_A s.t. $C_A^{\uparrow} := C_A^{\uparrow} \wedge (v_p = 3)$, i.e. the current position of the product has to be Workstation 3 for process operation O_A to be satisfied. The position variable for the mapping between the virtual resources and the formal SP model is denoted v_p. Operation O_U, which is realized either in Workstation 1 or 2 is extended with a condition $(v_p = 1) \vee (v_p = 2)$.

Besides operations for refining a product, a manufacturing system needs operations for transporting the products between different positions. A *transport operation* models transport of a product between positions in an FMS. A transport operation is *feasible* if it models a transportation that is physically realizable in the FMS.

The transport planning problem may now be expressed in the three iterative steps: to identify feasible transport operations, define their execution constraints and to guarantee a correct execution behavior. The transport planning should result in a correct global behavior. This is formalized in a criterion, not to execute transport operations without cause.

Criterion 3. (Criterion for transport planning). 3a) transport operations should only be enabled to workstations where the product has non-executed process operations. 3b) transport operations should only be enabled from workstations when the process operation has been executed. 3c) the number of operations between two process operations are limited to either one transport operation between two workstations or two transport operations between workstation to buffer and then buffer to workstation. □

The positions where the robot can pick and place a product (the movable fixture) have to be defined in order to generate feasible transport operations. A transport operation from a position i to a position j is termed O_{ij}. The plug-in in PS (Sundström, 2010) enables a user to choose the movable fixture of a certain product and to specify in which workstation(s) and buffer(s) the product may be placed. The workstations earlier specified for realization of process operations are automatically selected. The procedure of selecting positions takes place in the virtual cell, see Figure 3. The user should also specify what mover(s) that are intended to move the product between the positions. The complete combinatorial setup of transport operations are automatically created when

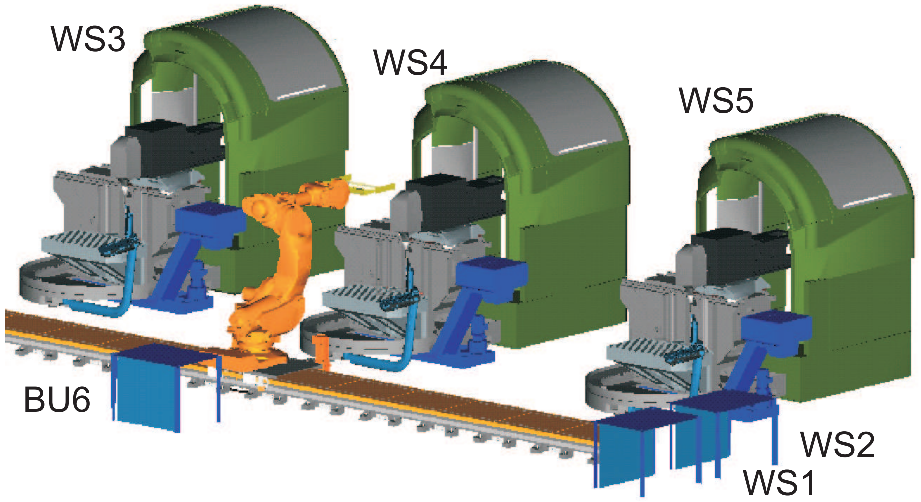

Fig. 3. The FMS in the example comprises six positions, five workstations (WS1-WS5) and one buffer (BU6). A robot moves products between positions. WS1 and WS2 are in- and output to cell and both can be used for (un-)fixation of a product to (from) a movable fixture. WS3, WS4, and WS5 are processing machines. The product in the example may be moved by the robot to all positions in the cell.

the positions and the mover(s) have been defined. A single mover M for n positions gives $n(n-1)$ transport operations to be realized by M. Reachability and collision tests may be included to guarantee that only feasible transport operations are created.

The position variable v_p may attain the values $v_p = 1, 2, 3, 4, 5, 6$ (five workstations and one buffer) for the movable fixture in the example. Each transportation between two positions is modeled as one transport operation. $6 \times 5 = 30$ transport operations exist since all position combinations are assumed feasible for the robot. The feasibility is reflected in two conditions for each transport operation. The first condition requires that the product is in position i. Hence, the pre-condition $C^{\uparrow}_{ij \mid i,j=1,\ldots,6\ i\neq j} := (v_p = i)$. The second condition updates v_p with an action when the movement is finished. Thus, the post-condition $C^{\downarrow}_{ij \mid i,j=1,\ldots,6\ i\neq j} := (v'_p = j)$. The feasible transport operations are generated to the model in SP.

A transport operation between Workstation 1 and 3 requires that the product has to be in Workstation 1, $v_p = 1$, for the transport operation to be satisfied. The pre-condition $C^{\uparrow}_{13}$ for operation O_{13} is $v_p = 1$. When the product is moved by the operation to Workstation 3, the variable is updated accordingly, $v'_p = 3$. This gives post-condition $C^{\downarrow}_{13} := (v'_p = 3)$.

The planning problem is now to plan the created transport operations based on the requirements given by Criterion 3. The complexity emerges since the enabling of each transport operation is based on the global state of the system, i.e. which process operations that have, or have not, been executed. The result is illustrated with a reachability graph, where process and transport operations represent the nodes. The reachability graph for the refinement process of the example product in the FMS is illustrated in Figure 5. The large number of possible sequences emphasize the use of a reliable planning method, neither to disable a satisfied transport operation nor to enable an unsatisfied transport operation in any state of the system. The possible blow up of the reachability graph may make it hard to overview and impractical to work with. An alternative way of presenting the planning result may therefore be desirable.

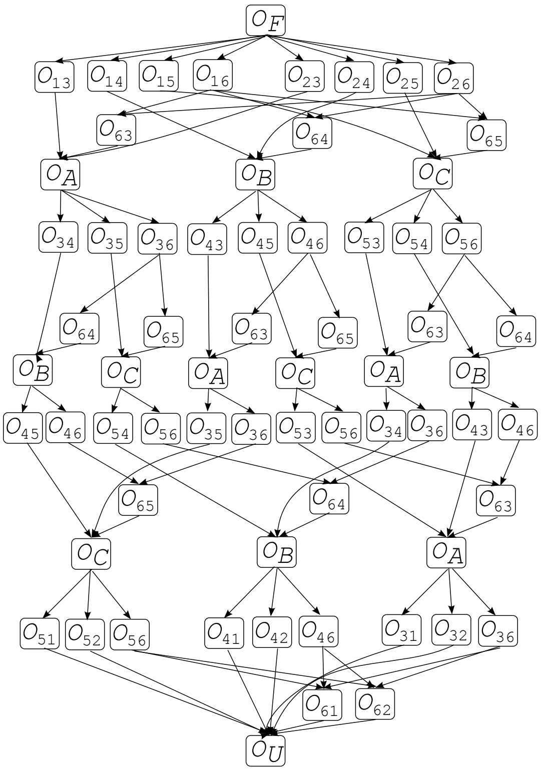

Fig. 5. Reachability graph for process and transport operations in the example. The graph presents the most flexible transport operation planning possible, with respect to Criterion 3.

5. SEQUENCE PLANNING

The idea of the planning method proposed in this paper is to transfer the transport operation planning problem into the problem of finding a non-blocking, deadlock free, solution for a discrete model of the product refinement. The approach is motivated through a series of observations. First of all, the solution to the transport planning is based on the sequential relations that exists between the process operations and what transport operations that are feasible. This solution should be bounded by Criterion 3. Secondly, the process and transport operations may be modeled as EFA models (Lennartson et al., 2010). Such a transformation to a discrete formulation enables employment of SCT (Ramadge and Wonham, 1989), and thereby calculation of a non-blocking supervisor, see for example Queiroz and Cury (2000) and Vahidi et al. (2006). A non-blocking supervisor prevents the start of operation sequences that are unable to reach some desirable location(s) in the discrete model. There exist methods to realize the calculated supervisor as an extension of the transition conditions in the existing EFA models (Miremadi et al., 2010). The EFA models may thereafter be re-transformed into operations. Thus, a desirable approach would now be to locally modify the EFA models and then use already existing algorithms to solve the planning problem and guarantee that Criterion 3 is fulfilled. A thorough description of the local modification is presented in what follows.

The operations, defined as presented in the former Section, are here described by EFA models. The term EFA model is prefixed with process or transport to stress the underlying operation type if needed. The initial SP model for the example will comprise: Five process EFA models (cf. Figure 4), 30 transport EFA models $E_{ij \mid i,j=1,\ldots,6,\ i\neq j}$ and the position variable for the product, $v_p = 1, 2, 3, 4, 5, 6$.

A transport operation may be executed more than once during the life-time of a product in an FMS, e.g. a transport operation from a buffer to a workstation that realizes multiple operations for a product. A reset transition is therefore introduced in each transport EFA model (Lennartson et al., 2010). No transition condition exists on the reset transition.

Criterion 3b, that a transport operation from a workstation should only be enabled when a process operation has been executed, and Criterion 3c, that there is an upper boundary for the number of transport operations that are allowed between two process operations, may be captured without introduction of SCT. Neither of the two criterions require any global knowledge of the system, i.e. which transport operations that are currently satisfied. It is enough to express the criterions locally in each operation directly. The transition conditions in the EFA models may be used for this. A new variable, v_t, is introduced in the SP model s.t. $v_t = a, p$. $v_t = a$ ($v_t = p$) means that transport operations are allowed (prohibited) to execute. Criterion 3b and 3c may then be expressed as

$$C_k^{\downarrow} := C_k^{\downarrow} \wedge (v_t' = a) \ s.t. \ k = A, B, C, F, U \qquad (1)$$

$$C_{ij}^{\uparrow} := C_{ij}^{\uparrow} \wedge (v_t = a) \wedge (v_t' = p) \ s.t \ i, j = 1, \ldots, 6 \ i \neq j \ (2)$$

in all process and transport operations respectively. (1) enables a new transport operation when a process operation has finished. (2) restricts any succeeding transport operations after the execution of itself. Transport operations to buffers will, however, lack the action $v_t' = p$ and thereby enable a successor buffer to workstation transport operation. $(v_p = i) \wedge (v_t = a)$ in $C_{i6}^{\uparrow}$ $s.t. i = 1, \ldots, 5$ in the example.

The non-blocking feature in SCT may now be applied to solve the planning problem with respect to the remaining criterion, Criterion 3a. The enabling of transport operations should be made with respect to what process operations that have already been executed. Non-blocking supervisor calculation is one way to take such global knowledge of the system into consideration.

Non-blocking calculation requires that marking is introduced in the SP model (Ramadge and Wonham, 1989), i.e. selection of some desirable location(s) to reach. A reasonable selection, in this context, is to enforce that all process operations are finished, i.e. that the product refinement is complete. This is achieved if the general marking, given for an EFA model in Definition 2, is changed for all process EFA models. A more suitable marking for the process EFA models in the example is therefore $M_k = O_k^f \ s.t. \ k = A, B, C, F, U$.

The initial EFA models in the example, without conditions on v_t, may lack blocking locations. All process operations will eventually be finished. The addition of v_t to the transition conditions for the EFA models will possibly introduce blocking locations. This is desirable, because v_t is not only added to fulfill Criterion 3b and 3c but also to generate blocking if a transport operation that breaks Criterion 3a is executed. Non-blocking supervisor calculation done according to Miremadi et al. (2010) will extend the requirements in the transition condition of each transport operation not to be satisfied in a state of the system where its execution will result in blocked locations. This shows the power of SCT; specify something locally and it is globally prevented before it happens.

The supervisor calculation in Miremadi et al. (2010) consists of a number of steps. The first step is to implicitly build up a global system model, S_0. States in S_0 from which no marked location can be reached through any sequence of events, are removed. The resulting system model, S, is the supervisor. The satisfiability for each operation is thereafter analyzed. States where an operation O_k is satisfied in S determine the specific values of variables and the locations that have to be active (or not) in other operations for O_k to be allowed to execute. States where an operation O_k is satisfied in S_0 but not in S, similarly determine when O_k can definitely *not* be executed since this would eventually block the system.

Let a situation in the example show how blocking appears. Assume the location set $\{O_F^f, O_A^f, O_B^e, O_C^i, O_U^i\}$. One transport operation will be enabled when the transition with label $O_B^{\downarrow}$ occurs, because $v_t' = a$ in $C_B^{\downarrow}$. $v_p = 4$ because process operation O_B is realized in Workstation 4. Assume now that a transport to Workstation 3, O_{43}, is executed, thus $v_t' = p$ in $C_{43}^{\uparrow}$ and $v_p' = 3$ in $C_{43}^{\downarrow}$. The single process operation, O_A, which is to be realized in Workstation 3 has already been finished. The overall system is now blocking. Process operation O_U occupies $O_U^i \neq O_U^f$. $v_t = p$ disables any new transport operation because $v_t = a$ in all $C_{3j \mid j=1,2,4,5,6}^{\uparrow}$. Supervisor calculation done according to Miremadi et al. (2010), for this situation, will result in $C_{43}^{\uparrow} := C_{43}^{\uparrow} \wedge O_A^i$.

The new conditions generated for the EFA models during the transport planning are finally added to the conditions in the user friendly operations. Hence, a solution to the transport operation planning problem with respect to Criterion 3 is available as operations after the supervisor calculation. The engineer may thereafter visualize the sequences of operations in different views. This way to represent the solution is hopefully easier to overview and simpler to implement than the complete operation reachability graph. Process operation O_A and transport operations to and from Workstation 3, where O_A is realized, are shown in Figure 6, as an example. Compare the readability of Figure 6 with the reachability graph in Figure 5.

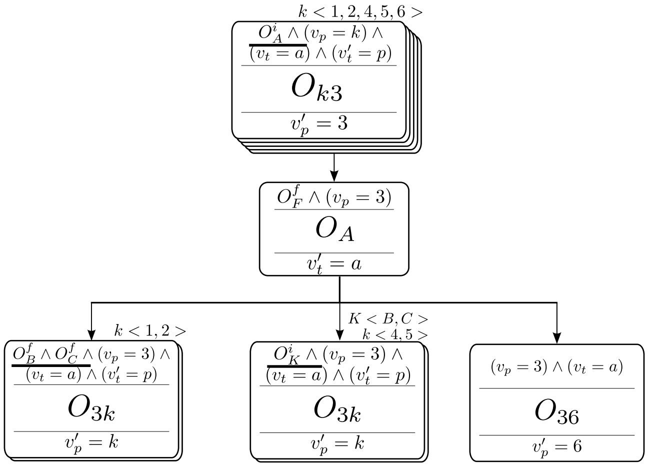
$k < 1, 2, 4, 5, 6 >$
$O^i_A \wedge (v_p = k) \wedge$
$(v_t = a) \wedge (v'_t = p)$
O_{k3}
$v'_p = 3$
$O^f_F \wedge (v_p = 3)$
O_A
$v'_t = a$
$k < 1, 2 >$
$O^f_B \wedge O^f_C \wedge (v_p = 3) \wedge$
$(v_t = a) \wedge (v'_t = p)$
O_{3k}
$v'_p = k$
$K < B, C >$
$k < 4, 5 >$
$O^i_K \wedge (v_p = 3) \wedge$
$(v_t = a) \wedge (v'_t = p)$
O_{3k}
$v'_p = k$
$(v_p = 3) \wedge (v_t = a)$
O_{36}
$v'_p = 6$

Fig. 6. Process operation O_A and transport operations to and from Workstation 3, where O_A is realized. The underlined pre-conditions are added as a result of the supervisor calculations for the planning problem. k and K are parameters that take one value in each instance. The operation O_{k3} exists e.g. in five instances.

6. CONCLUSION

A working method to automate transport planning for flexible manufacturing systems (FMSs) has been presented. The method enables specification of product requirements and FMS transport capabilities within standard virtual manufacturing tools. The requirements and capabilities are automatically translated to formal models. Non-blocking supervisory control is employed in order to take the full production situation for the FMS into consideration. The planning result may be visualized in the graphical SOP-language. Future work will focus on transport planning for parallel product refinement and how to extend the use of standard virtual manufacturing tools for automatic creation of formal models. Parallel product refinement requires EFA models that allocate and deallocate resources in the FMS.

REFERENCES

A. Adlemo, S.-A. Andreasson, M. Fabian, P. Gullander, and B. Lennartson. Towards a truly flexible manufacturing system. *Control Engineering Practice*, 3(4):545–554, 1995.

K. Bengtsson. *Operation Specification for Sequence Planning and Automation Design.* Licentiate thesis, Signals and Systems, Chalmers, Göteborg, Sweden, 2009.

J. A Buzacott and D. D Yao. Flexible manufacturing systems: a review of analytical models. *Managment Science*, 32(7):890–905, 1986.

George Chryssolouris. *Manufacturing Systems: Theory and Practice.* Springer, Mechanical Engineering Series, 2006. ISBN 9780387256832.

Panagiotis Kouvelis. Design and planning problems in flexible manufacturing systems: a critical review. *Journal of Intelligent Manufacturing*, 3(2):75–99, 1992.

B. Lennartson, K. Bengtsson, C. Yuan, K. Andersson, M. Fabian, P. Falkman, and K. Åkesson. Sequence planning for integrated product, process and automation design. *IEEE Transactions on Automation Science and Engineering*, 7(4):791–802, 2010.

Zhi Wu Li, Meng Chu Zhou, and Nai Qi Wu. A survey and comparison of petri net-based deadlock prevention policies for flexible manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 38(2):173–188, 2008.

S. Miremadi, K. Åkesson, B. Lennartson, and M. Fabian. Supervisor computation and representation: A case study. In *Proc. 10th International Workshop on Discrete Event Systems, WODES´10*, Berlin, Germany, 2010.

Michael Pinedo. *Planning and Scheduling in Manufacturing and Services, 2nd ed.* Springer, Series in Operations Research and Financial Engineering, 2009. ISBN 9781441909091.

Process Simulate. "http://www.plm.automation.siemens.com", 2010.

Max H. de Queiroz and José Eduardo Ribeiro Cury. Modular supervisory control of large scale discrete event systems. In R. Boel and G. Stremersch, editors, *Discrete Event Systems, Analysis and Control*, pages 103–110. Kluwer, 2000.

P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, JAN 1989.

M. Sköldstam, K. Åkesson, and M. Fabian. Modelling of discrete event systems using finite automata with variables. In *Proc. 46th IEEE Conference on Decision and Control*, New Orleans, USA, Dec 2007.

K. E. Stecke. Design, planning, scheduling, and control problems of flexible manufacturing systems. *Annals of Operations Research*, 3(1):1–12, 1985.

N. Sundström. Automatic generation of operations for the flexa production system. Master's thesis, Chalmers University of Technology, 2010.

A. Vahidi, M. Fabian, and B. Lennartson. Efficient supervisory synthesis of large systems. *Control Engineering Practice*, 14(10):1157–1167, October 2006.