Process Centred versus Resource Centred Modelling for Flexible Production Lines

Claudia Fiedler and Wolfgang Meyer[1]

Abstract—**We describe a resource oriented modelling method for robotic flowshops and examplify it on a galvanic plant. We compare the process oriented modelling method with the resource oriented method. The resulting simulation tool can be used for the design of scheduling algorithms. Solutions can be found to compromise between the use of resources and productivity of the plant.**

Index Terms—**Timed petri nets, hoist scheduling, flexible manufacturing**

I. Problem Statement

The increasing use of flexible production environments poses high demands on production planners. Besides the necessity to optimize the stationary production over a long period it is more and more important to be able to change quickly and efficiently between different product modes. For plants with automated transport systems we have to find optimum control sequences for the transporter to meet the requirements. Therefore we have developed a simulation model to find control sequences both for stationary and for flexible production environments.

The application considered is a line of basins containing chemical, electrolytic or rinsing bathes served by one or more transporters. The plant consists of m machines $M_1, \ldots, M_m$, an input station M_0 and an output station M_{m+1} sometimes combined at the same place. The input station contains a set of parts J. Each part has to be processed according to its process plan, the list of the operation times o_i, $i \in M$ at the machines and the transport times t_{ij}, $i, j \in M$ between them.

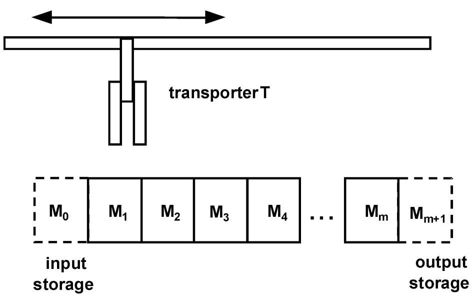

Fig. 1 Layout of the plant

[1] Claudia Fiedler and Wolfgang Meyer are with the Institute of Automation, Hamburg University of Technology, 21071 Hamburg, Germany
(e-mail: claudia.fiedler@tu-harburg.de; w.meyer@tu-harburg.de)

The operation times o_i of part J are kept in intervals $\left[l_i^J, u_i^J\right]$ with a lower bound l_i^J and an upper bound u_i^J . If the upper bound is equal to the lower bound, we speek of a no-wait condition. The upper bound can be infinity, too. There are one or more transporters T_n concurrently or operating with defined areas on the same or on different tracks. The travel times δ_{ij} can be constant, additive or Euclidean. Additive travel times follow the triangle equality and Euclidean travel times follow the triangle inequality. They are symmetric ($\delta_{ij}=\delta_{ji}$) and zero from a machine to itself ($\delta_{ii}=0$). The transport times t_{ij} between the operations o_i are the sum of travel times δ_{ij} and a constant needed for loading and unloading the part. The parts in the input station can be of the same type or of different types. Depending on the types of the parts the goal is either to minimize the cycle time v_i for parts of the same type or to minimize the the throughput time in case of different part types.

II. State of the Art of Scheduling

The general problem is known as robotic flowshop scheduling. The part input sequence (for different parts in the input buffer) has to be specified as well as the sequence of robot moves. [1]and [2] are recommended to get a general idea. We adress the Hoist Scheduling Problem as a special case of robotic flowshops. The operation times are given in intervals. The transporters have Euclidean travel times and loaded transporters are not allowed to wait. The NP-completeness is proven by Crama and Klundert [3]. Phillips and Unger [4] solved the monocyclic case with integer programming. Rodozek and Wallace used a hybrid constraint logic programming (CLP) and mixed integer programming (MIP) algorithm [5]. An overview over different kinds of hoist scheduling problems is given in [6]. They extend the Graham notation applied to robotic flowshop scheduling [2][7] to the varying problems of hoist scheduling.

III. Processbased Modelling

In [8] we presented a process centred modelling method according to the modeling method of [9] for the cyclic hoist scheduling. The general structure is shown in Fig. 2. The parts modelled as process tokens with the processing times as attributes are released to the request generator with constant release time v. The model of the plant (request generator) sends transport requests to the request sequencer if the state of the model has changed because of a finished transport operation and the starting of a tank operation. According to the given priority, the sequencer decides which of the transport request is

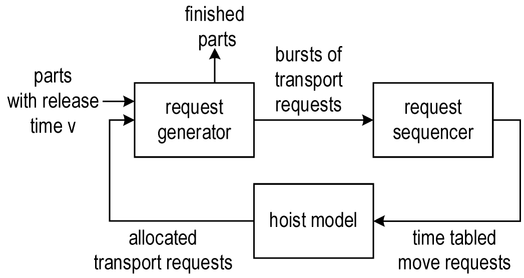

Fig. 2 Scheduling model

answered next when the transporter is available. Then the time tabled request releases a transporter move if the transporter is not at the needed place. The allocated transporter then causes a transport operation and a new transport request. There are enough process token to lead to a stationary behavior after a transient region at the beginning with the suitable release time V. The start value of V is the sum of the maximal operation time in a tank and the transport times to and from the tank. If a cyclic behavior can not be reached or if the operation times exceed the upper bounds of the given intervals, the release time is increased and the simulation starts again until the given constraints are fulfilled.

In the request generator the tank and transport operations of a job are lined up according to the process plan (Fig.3).

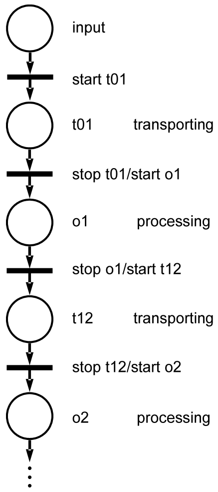

Fig. 3 Process plan as A-path

Each operation begins and finishes with a start/stop-transition. This is the so-called A-path. Then the B-path is added: the needed resources for the operations connnected with the start transition of the correspondent operation. In our example we need tank 1 and tank 2 for the tank operations and the transporter for the transport operations. This may vary if there are more transporters or loops in the process plan if a tank is used more than once for a job. Process token symbolize the parts and resource token for the availability of resources are added. A tank resource is occupied if a transport operation to the tank has started and as long the transport operation from the resource has not been finished (Fig.4). The requests are collected in the input place for the sequencer. It is

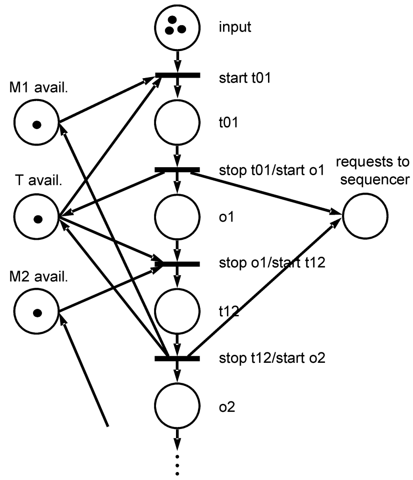

Fig. 4 A-path and B-path

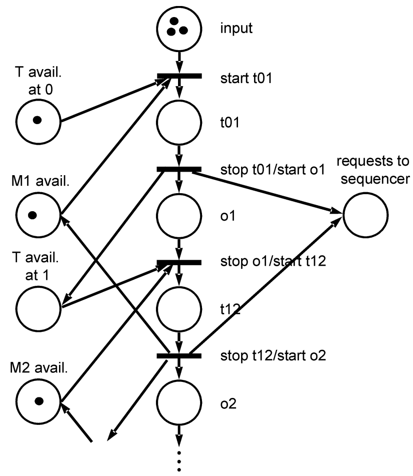

Fig. 5 Request generator

not only important if the transporter is available but also if it is at the needed place. Therefore the transporter availability place is extended to places for the availability at the needed tank (Fig. 5).

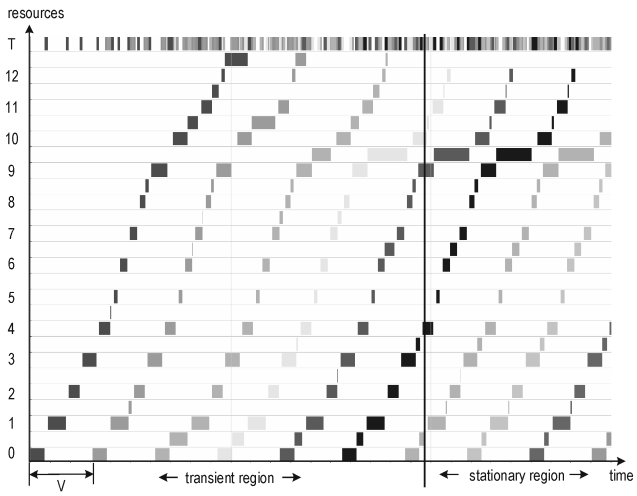

Fig. 6 Gannt Chart for solution of PhU-benchmark - transient and stationary regions

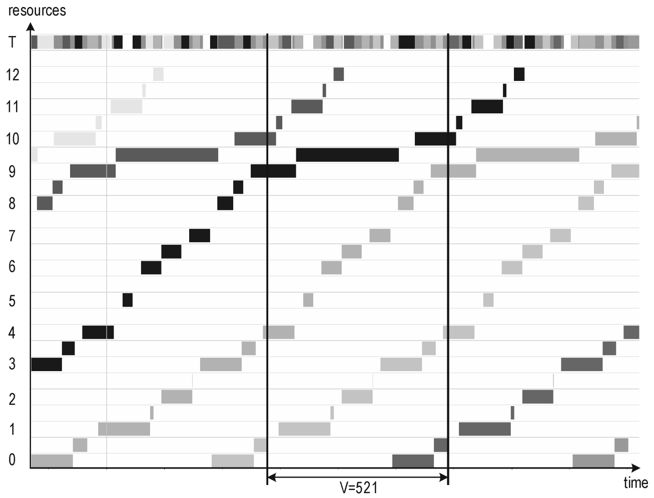

Fig. 7 Gannt Chart for solution of PhU-benchmark – transporter sequence

Because the loaded transport operations are included in the request generator the transporter model just contains the unloaded movements from each to each other place. Applied to the first benchmark described by Phillips and Unger (PhU) [4] the modelling leads to the optimal solution of 521 seconds for the cycle time V [8]. Fig. 6 shows the gannt chart of the minimal solution of the Phillips/ Unger benchmark problem. The processes are released to the plant with a release time V of 521 seconds and the operation times are given as the lower bounds of the intervals. The rows between two resources symbolise the extension of the operation time from the lower bound. If there is more than one part in the plant the sequencer descides according to the implemented rule the order of loaded and unloaded transport operations for the transporter T as the bottleneck resource.

The descision time of the sequencer is time shifted by the maximal value of the movement time for unloaded transports to enable the transporter to be at the needed tank in due time. The implemented rule here is a special priority rule depending on the size of the operation intervals as described in [8]. The Operation-Due-Date-rule (ODD) wich chooses the next tank dependent on the time difference to the upper bound of the interval leads to good results, too. The more parts are in the plant the more the operation times are extended. After the transient region the transporter shows stationary behavior in the minimal time interval of the release time V=521 seconds. Fig. 7 shows the transporter sequence with the length V. The sequence can be transferred into a programmable logic controller (PLC) to realise the processes with the given constraints in the real plant.

The PhU problem is modelled as a flowline. Therefore deadlocks can not take place. If there are loops in the process plans deadlocks can occur and an deadlock avoidance algorithm has to be implemented. Possibilities to inhibit deadlocks are described in [10]. Our approach is described in chapter V.

IV. RESOURCE CENTRED MODELLING

In flexible manufacturing environments there is a fast change in product types. To find sequences for lot switching or for new products the process centred model is unsuitable because for each new

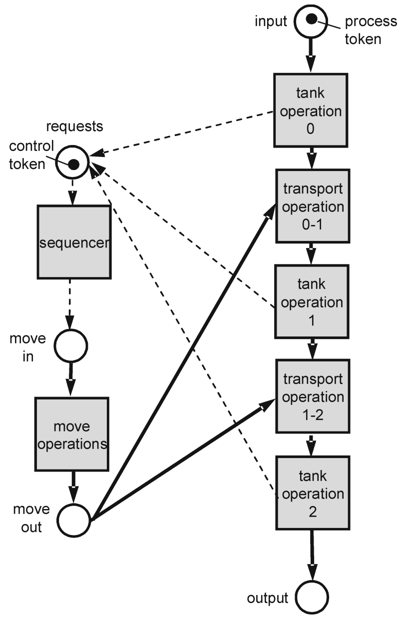

Fig. 8 Process centred model (A-path model)

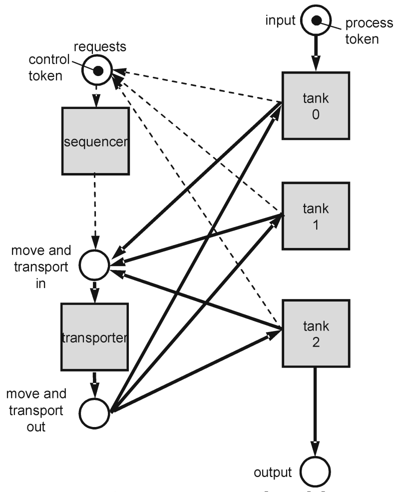

Fig. 9 Resource centred model

process plan a new A-path has to be implemented. In Fig.8 the simplified process centred model is shown. The A-path as the sequence of tank and transport operations for a process are on the right and the move operations and the sequencer are on the left side. The signal flow of the requests takes place along the dashed lines the process flow occurs along the bold lines. Just the operation times can be changed by input data not the sequence of operations. In Fig. 9 the resource centred model is shown. Compared to the process centred model not the operations but the resources are modeled. The signal flow is similar to the process centred model but the process flow is composed of single operation elements using the corresponding resources. Therewith flexible A-pathes are possible. The flexibility is reflected in Fig. 9 in the number of process flow connections, too. In the process centred model there is just one way for the parts whereas in the resource centred model the processes can be composed in any order. In [11] the compact modeling as a similar concept is described and the effects on the number of petri net elements are determined.

V. Deadlock avoidance

In Discrete Event Systems with loops in the process plans deadlocks may occur. We therefore need a deadlock avoidance algorithm in the resource centred model to enable the model to simulate processes with loops in the plan. The idea is to prevent the last change in the state of the plant which closes a deadlock. The following example may illustrate the algorithm:

Given are three resources a_1, a_2 and a_3. Each resource has capacity one. Then each resource can handle just one of the processes. For process P1 the actual resource may be a_1, for process P2 a_2 and for P3 a_3. Then these four possibilities for the following two resouces for the three processes are possible:

$$\text{P1}=\begin{pmatrix}(a_1a_2a_1)\\(a_1a_3a_1)\\(a_1a_2a_3)\\(a_1a_3a_2)\end{pmatrix};\ \text{P2}=\begin{pmatrix}(a_2a_1a_2)\\(a_2a_3a_2)\\(a_2a_1a_3)\\(a_2a_3a_1)\end{pmatrix}\ \text{and P3}=\begin{pmatrix}(a_3a_1a_3)\\(a_3a_2a_3)\\(a_3a_1a_2)\\(a_3a_2a_1)\end{pmatrix}.$$

Each combination of the three processes P1, P2 and P3 is a deadlock with either size 2 if two processes or resources are involved or size 3 for three involved processes. For example, if P1=(a_1 a_2 a_1) and P2=(a_2 a_1 a_2) then there is a deadlock because P1 blocks the next tank of P2 and P2 uses the next resource of P1. A deadlock with size 3 occurs if P1=(a_1 a_2 a_1), P2=(a_2 a_3 a_2) and P3=(a_3 a_1 a_3) because there are three involved processes and for each of the three processes there exist a process which uses the next resource.
That means there is a deadlock if :

Let **P** be the set of processes in the plant

$$\mathbf{P}=\{P1\ldots Pn\};\quad n\in\mathbb{N} \tag{1}$$

and each P_i consists of the actual and the next operation,

$$P_i=\left(a^i_{actual}\ a^i_{next}\right);\ i=1\ldots n \tag{2}$$

then there exists a subset **Q** of **P**

$$\mathbf{Q}\subseteq\mathbf{P};\ \mathbf{Q}=\{Q_1\ldots Q_m\};\ m\in\mathbb{N} \tag{3}$$

with

$$\mathbf{A}^{\mathbf{Q}}_{\mathbf{actual}}=\mathbf{A}^{\mathbf{Q}}_{\mathbf{next}} \tag{4}$$

where $\mathbf{A}^{\mathbf{Q}}_{\mathbf{actual}}$ is the set of the actual resources occupied by the processes of **Q** and $\mathbf{A}^{\mathbf{Q}}_{\mathbf{next}}$ the set of the next resources used by the processes of **Q**.

In the implementation of the resource centred model then we have to prohibit the transport of the last part to that resource which leads to (4) and results in a deadlock.

In Fig. 10 the simplified model with deadlock avoidance is displayed. The timetabled request is send from the sequencer to the deadlock avoidance modul with information about the actual and the next two resources. The algorithm descides if a deadlock occurs if the process will be transported to the next tank. If so the deadlock avoidance modul sends an inhibit signal to the sequencer for this process and tries the next one. Every time the state of the request generator will change caused by a transport operation all the inhibited requests stored in the deadlock avoidance modul are tested whether the danger for deadlock still holds or not. The sequence decision for concurrent processes is based on the due date of the tank operation the transport would finish. Other priority decisions are possible, too.

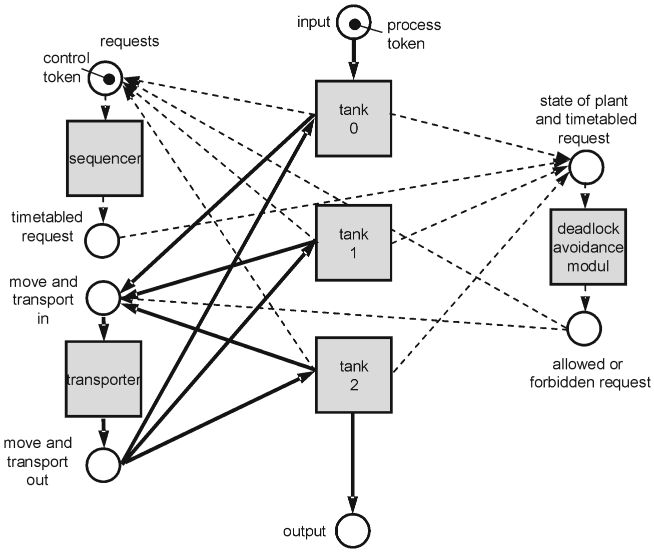

Fig. 10 Resource centred model with deadlock avoidance module

VI. Illustrative example

The implemented simulation tool can be used for scheduling algorithms decisions. The result is a transporter sequence in a textfile which can be directly transfered into a PLC to control the transporter. The input is an Excel file with the process plan and the transporter road map. The model is implemented in PACE 5.0, a simulation tool for coloured timed petri nets. [12] Assume the process plan as given in table 1. There are seven tanks.

Table 1. Process plan P1

resources	M0	M1	M2	M3	M4	M5	M6	M0
lower bounds	120	400	90	200	100	130	100	0
upper bounds	150	460	120	230	160	155	125	1000
transport times	move times from tab 2 plus 20							

The input station equals the output station. The operation times are given in lower and upper bounds. The unloaded move times of the transporter have been taken from Phillips Unger benchmark and are given in table 2. The loaded transport times between the tanks have been calculated from the unloaded move times plus 20 time units.

Table 2. Unloaded Moves of the Transporter for PU12 Benchmark

from/to	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	11	14	16	14	19	22	24	26	29	6	8	10
1	11	0	2	5	2	8	10	13	15	17	10	3	1
2	14	2	0	2	0	5	8	10	13	15	12	6	3
3	16	5	2	0	2	3	5	8	10	13	15	8	6
4	14	2	0	2	0	5	8	10	13	15	12	6	3
5	19	8	5	3	5	0	3	5	7	10	18	11	9
6	22	10	8	5	8	3	0	2	5	7	20	14	11
7	24	13	10	8	10	5	2	0	2	5	23	16	14
8	26	15	13	10	13	7	5	2	0	2	25	19	16
9	29	17	15	13	15	10	7	5	2	0	27	21	19
10	6	10	12	15	12	18	20	23	25	27	0	7	9
11	8	3	6	8	6	11	14	16	19	21	7	0	2
12	10	1	3	6	3	9	11	14	16	19	9	2	0

The result for the flowline is given in Fig. 11. The minimal stationary solution is V=489.

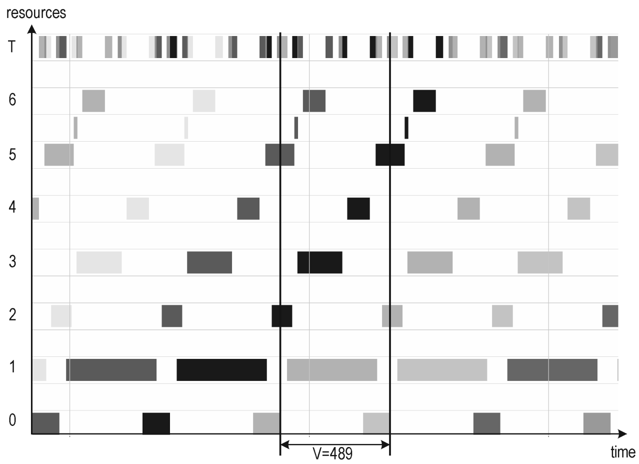

Fig. 11 Gannt chart for the example in table 1

The operation in tank 1 seems to be the bottleneck. If we add another resource M1 in the line the result can be reduced by 41% to V=292 (Fig. 12). Now the transporter is fully occupied and it is unlikely to find a smaller solution without adding another transporter.

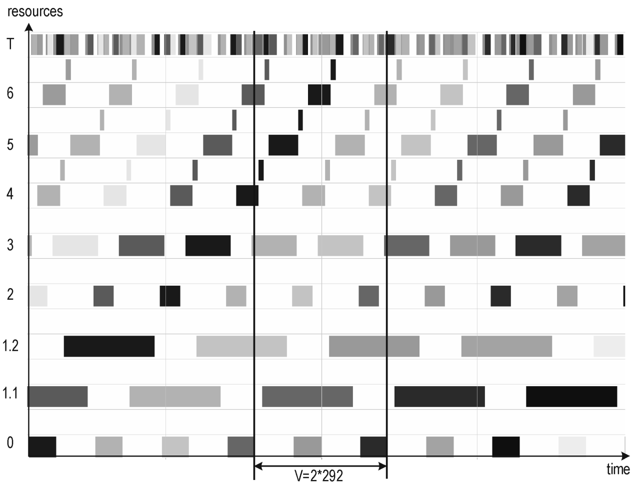

Fig. 12 Gannt chart for the example in table 1 with an additional tank 1

If the operation in tank 2 is a rinsing operation and the operation in tank 5 too, we can test what happens if we use just one tank for this operation. Fig. 13 shows the result for just one tank 1 and the loop in the process plan for tank 2. The result of V=801 is really poor. Because of the loop, the smaller stationary solutions found before exceed the upper bounds of the operation times.

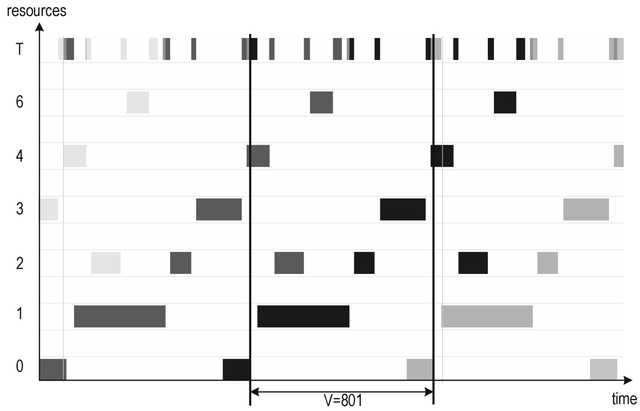

Fig. 13 Gannt chart for the example in table 1 with M2=M5

But if we add another tank 1 we find a solution of V=385 as the best compromise between the number of resources and the productivity (Fig. 14).

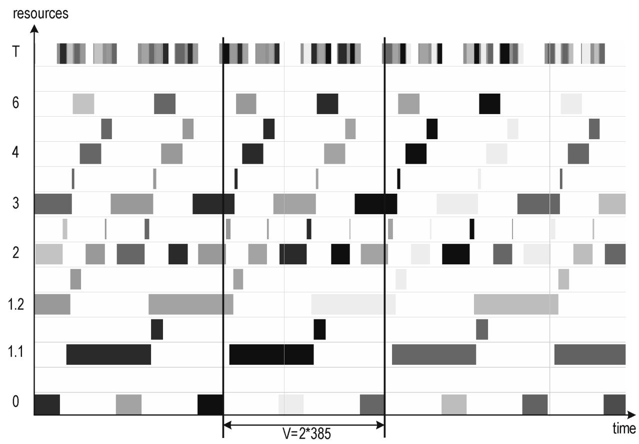

Fig. 14 Gannt chart for the example in table 1 with two M1 and M2=M5

VII. Conclusions

We described a modelling method for a simulation tool based on coloured timed Petri nets. It is possible to find stationary transporter sequences to feed them into a PLC to control the transporter. The process plans can be easily changed just by modifying an Excel file. The danger of deadlocks if there are loops in the process plans is resolved with a deadlock avoidance algorithm. Several plant layout scenarios can be tested to find the best compromise between use of resources and productivity. Lot switching solutions to minimise the time between two diffenrent products can be obtained, too. The tool is implemented for a real 32 tank plant with two transporters in a factory for electronic devices.

REFERENCES

[1] Y. Crama, V. Kats, J. van de Klundert and E. Levner, "Cyclic scheduling in robotic flowshops" in *Annals of Operations Research*, vol. 96, pp. 97–124, 2000.

[2] M. Dawande, H.N. Geismar, S.P. Sethi and C. Sriskandarajah, *Sequencing and Scheduling in Robotic Cells: Recent Developments* in *Journal of Scheduling*,.vol. 8, pp. 387–426, 2005.

[3] Y. Crama and J. Klundert, "Robotic flowshop scheduling is strongly NP-complete", in *Ten Years LNMB (W.K.Klein Haneveld, O.J. Vrieze and L.C.M. Kallenberg, eds.),* CWI Tract 122, Amsterdam, pp. 277-286, 1997.

[4] L. W. Phillips and P. S. Unger, "Mathematical Programming Solution of a Hoist Scheduling Program", *AIIE Transactions*, vol. 28, no. 2, pp. 219-225, June 1976.

[5] R. Rodošek and M. Wallace, "A Generic Model and Hybrid Algorithm for Hoist Scheduling Problems", *Lecture Notes in Computer Science*, vol.1520, pp. 385-399, Springer Verlag, Berlin, 1998.

[6] M.-A. Manier, C. Bloch, "A Classification for Hoist Scheduling Problems", *International Journal of Flexible Manufacturing Systems*, vol. 15, no.1, pp. 37 – 55, Jan 2003.

[7] R.L.Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnoy Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey", *Annals of Discrete Mathematics*, 5, 287-326, 1979.

[8] C. Fiedler, "Event-driven Generation of Periodic Hoist Schedules", *Proc. IEEE Conf. Systems, Man, and Cybernetics*, Taipei, Oct. 2006.

[9] M. Zhou and F. DiCesare, Petri Net Synthesis for Discrete Event Control of Manufacturing Systems, London: Kluwer Academic Publishers, 1993.

[10] M. Lawley, S. Reveliotis, and P. Ferreira, "Design Guidelines for Deadlock-Handling Startegies in Flexible Manufacturing Systems", *The International Journal of Flexible Manufacturing Systems*, Kluwer Academic Publishers, Boston, 9, pp. 5-30, 1997.

[11] A. v.Drathen, "Compact Modeling of Manufacturing Systems with Petri nets", *Proc. IEEE Conf. Systems, Man, and Cybernetics*, Montreal, Oct. 2007.

[12] www.ibepace.com