Analysis of Place/Transition Nets with Timed Arcs and its Application to Batch Process Control

Hans-Michael Hanisch

Universität Dortmund, Fachbereich Chemietechnik, Lehrstuhl Anlagensteuerungstechnik
Postfach 500 500, D-44221 Dortmund
Germany

Abstract. The paper presents an analysis method for Place/Transition nets with timing of arcs directing from places to transitions. Based on this class of timed nets, the corresponding state graph, called dynamic graph, and a method to compute the state graph are defined.
By means of the dynamic graph, the complete dynamic behaviour of the modeled system can be studied, objective functions can be formulated, and optimal control strategies can be computed. The concept is applied to two different problems of supervisory control of batch plants in the chemical industry.

Keywords

Timed Petri nets, system design, analysis, performance evaluation, batch process control

1 Introduction

Batch production systems usually consist of several process units (reaction, distillation, drying, filtering etc.) which perform sequences of technological operations and are coupled by flows of mass or energy and shared resources. The coupling pattern is time-dependent and many of the control actions are of a discrete nature such as opening/closing valves, starting/stopping pumps or agitators etc. Thus, discrete models must be applied to batch process systems.
Although the application of Petri nets to batch processes is still very sparse [1-6, 14-16], it has been shown by several applications which are summarized in [2] that Petri nets are an excellent tool to model and analyse the interactions between the process units in a batch plant and to describe the desired mode of operation of the plant as a basis for supervisory controller design.
The method presented here is based on a control and modeling hierarchy. Each single process in its single equipment is controlled by a local sequential controller and runs independently from other processes with the exception of interactions by

- flows of mass or energy from one process unit to another one and
- allocation of limited, shared resources (auxiliary devices, such as metering

and storage tanks, mains for steam and vacuum etc., which are temporarily used by the process units).

It is supposed that the local controllers ensure that the behaviour of the process units correspond to the technological requirements.

The *goal* of the modeling and analysis process is *not* to model the continuos chemical processes in the process units together with discrete control processes by means of hybrid models [8]. We suppose that the continuous processes are controlled by the single controllers for each process unit.

The *supervisory control* has to ensure that

- the flows of mass and energy between the single processes are performed according to the desired mode of operation,
- overloadings of limited resources are prevented, and
- an objective function (throughput, costs, or other) is optimized.

In order to achieve such a global behaviour, the supervisory control starts or delays sequences of operations in the process units and allocates shared resources to the process units.

Models of the desired behaviour require that a control strategy (i.e. the resolution of all conflicts, see Section 4) is pre-determined. In some cases, however, no initial knowledge about a feasible control strategy is available. Only a goal (throughput maximization, minimization of startup or shutdown times) is given which is to be achieved by the control. A concept to solve such problems must ensure that a suitable (or optimal) control strategy can be computed *as a result* of a modelling and analysis process. The modeling capability of causal, classical Petri nets and the corresponding analysis methods [12] are not sufficient since the goals which are to be achieved by the control explicitly include process times. Hence, timed Petri nets must be applied.

The most common concept of timed Petri nets is that of timed transitions [11], where each transition has a firing duration greater than zero. However, the models must be able to describe disturbances which can occur while a transition is firing and can cause states which are different from the states normally reached when the firing is finished. After a short period of using nets with timed places [1,14], the concept of timed arcs was chosen, which is described below. It has larger modeling capabilities than timed transitions or places (which are included as a special case of timed arcs) and is well suited for the modeling of batch processes as well as for other timed systems. By means of an appropriate software tool [13], models which are larger than the examples in Section 5 can be analysed.

2 Place/Transition Nets with Timed Arcs

A Place/Transition net with timed arcs is a tuple ATPN = (P,T,F,V,m_0,ZB,D_0) where (P,T,F,V,m_0) is a (causal) Place/Transition net [12].
For each $t \in T$ two |P|-dimensional vectors t^+ and t^- are defined which describe the change of the marking when t fires:

$$t^-(p) = \begin{cases} V(p,t), & \text{if } p \in Ft \\ 0, & \text{otherwise} \end{cases} \qquad t^+(p) = \begin{cases} V(t,p), & \text{if } p \in tF \\ 0, & \text{otherwise .} \end{cases} \tag{1}$$

Each arc (p,t) from a place to a transition is labeled with a time interval ZB(p,t):

$$ZB: (P \times T) \cap F \rightarrow \mathbf{N} \times (\mathbf{N} \cup \{\infty\}) \tag{2}$$

that describes the *permeability* of this arc. The permeability of an arc $ZB=[ZB_r,ZB_l]$ is a tuple of two integer values, where the first value $ZB_r(p,t)$ denotes the begin of the permeability and the second one $ZB_l(p,t)$ denotes the end of the permeability of the arc relative to the local clock of p.
The *state* z=(m,D) of a timed net consists of two components, the *marking* m of the net and *local clocks* D denoting the age of the marking of the places:

$$D: P \rightarrow \mathbf{N}. \tag{3}$$

It depends on the marking *and* the local clocks whether a transition is enabled or not. *A transition t is enabled* if all places $p \in Ft$ carry enough tokens according to the token weight of the arc and if all arcs directing to the transition are permeable. For each transition enabled at the marking m, we can compute the earliest firing time eft(t), which is the time the last pre(t)-arc becomes permeable:

$$eft(t) = \mathrm{Max}\ \{\ sgn(ZB_r(p,t)\ -\ D(p))\,|\,p \in Ft\ \} \tag{4}$$

$$sgn(i) := \begin{cases} i\ , & \text{if } i>0 \\ 0\ , & \text{otherwise.} \end{cases} \tag{5}$$

A transition, however, can only fire if no pre(t)-arc becomes unpermeable before the last pre(t)-arc becomes permeable. The time lft(t) when the first pre(t)-arc of transition t becomes unpermeable can be computed as follows:

$$lft(t) = \mathrm{Min}\ \{\ ZB_l(p,t)-D(p)\,|\,p \in Ft\}\ . \tag{6}$$

Note that eft(t) and lft(t) depend on the local clocks of the places $p \in Ft$.

The set K of transitions enabled in state z is given by:

$$K = \{ t \mid t^- \leq m \wedge eft(t) \leq lft(t) \} . \tag{7}$$

We use the earliest and maximal firing rule.
The *step delay* $\Delta\tau$ denotes the earliest time a transition can fire:

$$\Delta\tau = \text{Min} \{ eft(t) \mid t \in K \} .$$

The elements of E are the enabled transitions which fire exactly after $\Delta\tau$:

$$E = \{ t \mid eft(t) = \Delta\tau \wedge t \in K \} . \tag{9}$$

The set E must be partitioned into maximal subsets of transitions which can fire simultaneously after $\Delta\tau$ time units. Such subsets are called *maximal steps* u^*. For each maximal step u^* the vector Δu^* of dimension $|P|$ denotes the change of marking by firing u^*:

$$\Delta u^* = \sum_{t \in u^*} t^+ - t^- . \tag{10}$$

The firing of a maximal step u^* after $\Delta\tau$ time units creates a new state z' (abbreviated: $z[(u^*,\Delta\tau)>z')$ by changing the marking *and* the local clocks of the timed net:

$$m' := m + \Delta u^* \tag{11}$$

$$D'(p) := \begin{cases} D(p) + \Delta\tau \text{ , if } m(p)>0 \wedge p \notin Fu^* \cup u^*F \\ \qquad\quad 0 \text{ , otherwise.} \end{cases} \tag{12}$$

Equation (12) describes the change of the clock positions. Each clock of a marked place which is not connected with an element of u^* is increased by $\Delta\tau$, since the marking of these places is not changed and the marking therefore becomes $\Delta\tau$ time units "older". The other clocks are put back to zero since the marking of the places is changed by firing u^*.
The concept includes the special cases:

1. no delay of permeability ($ZB_r(p,t)=0$)
2. no limitation of permeability ($ZB_l(p,t)=\infty$).

If all arcs have a time interval $ZB=[0..\infty]$, the net behaves like a causal Petri net under the maximal firing rule.

3 The State Graph for Place/Transition Nets with Timed Arcs

Similar to the definition of the reachability graph of classical, causal Petri nets [12], we can define the *state graph* for arc-timed Place/Transition nets as follows.
Let G be the set of all state transitions:

$$G \subseteq \wp(T) \times \mathbf{N}\,, \quad \wp(T)\text{...Power Set of } T\,. \tag{13}$$

The abbreviation z[g>z' denotes that state z' can be reached from state z by state transition g, where g is a tuple of the maximal step u^* and the step delay $\Delta\tau$.
The set of all possible firing sequences of state transitions g is the set of words $W_{ATPN}(G)$. We call a state z' *reachable from state* z (abbreviated: z[*>z') if there is a sequence $q = g_1, g_2, \ldots, g_n$ with

- q is a word
- q transforms state z into state z':

$$z[*>z' :\Leftrightarrow \exists q\ (q \in W_{ATPN}(G) \wedge z[q>z')\,. \tag{14}$$

Let $R_{ATPN}(z)$ be the set of all states in the arc-timed Petri net which are reachable from state z. We can then define the state graph (dynamic graph) of an arc-timed Petri net as $DG_{ATPN}=(R_{ATPN}(z_0), A_{ATPN})$ with

$$A_{ATPN} = \{\, (z,g,z') \mid z,z' \in R_{ATPN}(z_0) \wedge g \in G \wedge z[g>z' \,\}\,. \tag{15}$$

Each element of A_{ATPN} describes an arc directing from node z to node z' and labeled with a maximal step $u^*(g)$ and the corresponding step delay $\Delta\tau(g)$.
In order to be able (at least potentially) to compute the dynamic graph for a given arc-timed Petri net, each of the two components of the state (the marking as well as the clock positions) must be bounded. Since the set of reachable markings (not states!) of a timed Petri net is always a subset of the reachable markings of the corresponding causal Petri net, any bounded causal Petri net with arbitrary time intervals of the arcs will have a finite number of markings.

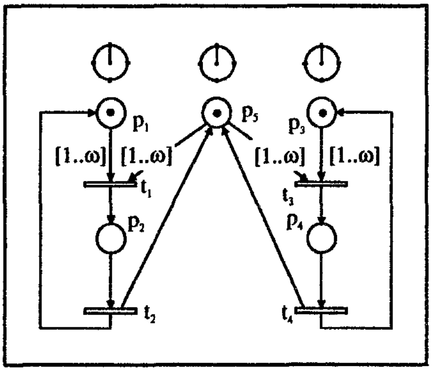

Fig. 1. Arc-timed Petri net with initial state

However, even for bounded Petri nets, the number of clock positions can be infinite, and some additional studies are necessary to prevent this. At first we give a small example of a net with an infinite number of clock positions and then describe some mechanisms to prevent infinite clock positions.
Fig. 1. shows a bounded (safe) Petri net with unbounded clock positions. The initial

state is given by the initial marking and by the clock positions (graphically represented by the clock symbols at places p_1, p_3 and p_5) which are all zero. Only the arcs (p_1,t_1), (p_5,t_1), (p_5,t_3), and (p_3,t_3) are labeled with the time interval $[1..\infty]$. Consequently, the arcs (p_2,t_2) and (p_4,t_4) have the time interval $[0..\infty]$.
Fig. 2. shows the corresponding dynamic graph with an infinite number of nodes caused by infinite clock positions. The nodes of the graph describe the state vectors $(m(p_1),D(p_1),...,m(p_5),D(p_5))$, where the marking durations for the marked places are given in parentheses. The maximal steps and - in parentheses - the corresponding step delays $\Delta\tau$ are given as arc labels.

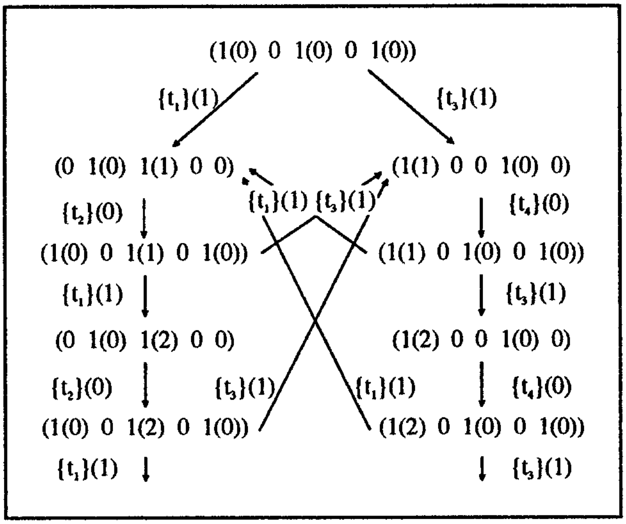

Fig. 2. Dynamic graph of the net of Fig. 1.

It can be seen easily that firing of t_1 and t_2 increases the clock position of p_3 by one. For an arbitrary long firing sequence which only consists of t_1 and t_2, the clock position of p_3 will exceed any bound. The same goes for a firing sequence of t_3 and t_4 and the local clock of p_1. On the other hand, values greater than one of the local clocks $D(p_1)$, $D(p_3)$ and $D(p_5)$ will obviously never influence the permeability of the corresponding arcs (after one time unit the arcs are permeable and stay permeable forever). Consequently, the clocks of places p_1, p_3 and p_5 do not need to indicate durations greater than one.
Therefore, we can modify the second part of the firing rule (12) as follows:

$$D'(p) := \begin{cases} D(p)+\Delta\tau \text{ , if } m(p)>0 \wedge p \notin Fu \cup uF \wedge D(p)+\Delta\tau \leq D_{max}(p) \\ D_{max}(p) \text{ , if } m(p)>0 \wedge p \notin Fu \cup uF \wedge D(p)+\Delta\tau > D_{max}(p) \\ 0 \text{ , otherwise .} \end{cases} \qquad (16)$$

where $D_{max}(p)$ is one for places p_1, p_3 and p_5 and zero for places p_2 and p_4 in the example of Fig. 1..
With the modified firing rule, the dynamic graph of Fig. 3. is computed which describes the same language as the dynamic graph in Fig. 2..
Based of the above idea, we can find for any place and arbitrary time inscriptions a finite bound $D_{max}(p)$ which denotes the maximal value of the local clock $D(p)$ that influences the permeability of the arcs (p,t), $t \in pF$ which are connected with p. We

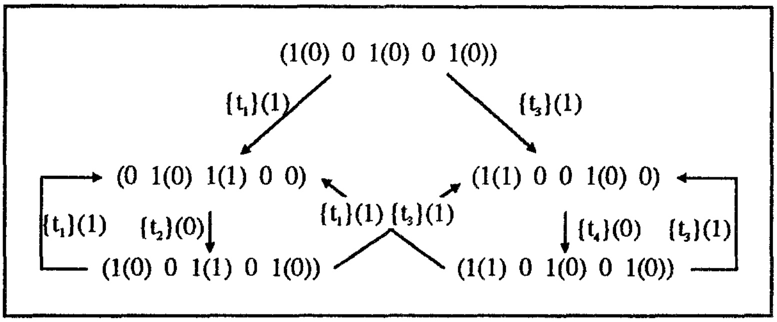

Fig. 3. Reduced dynamic graph

have to distinguish three different cases:

1. All arcs (p,t) which are connected with place p are labeled with [0..∞]. In this case place p does not need a clock at all. Hence, $D_{max}(p)$ is zero:

$$D_{max}(p) = 0 \Leftrightarrow \forall\ t \in pF\ (ZB(p,t) = [0..\infty])\ . \tag{17}$$

2. The maximal finite time inscription of all arcs (p,t) denotes the *begin* of the permeability of an arc. Then $D_{max}(p)$ is equal to this value:

$$\begin{aligned} ZB_{rMax}(p) &= \mathrm{Max}\ \{ZB_r(p,t)\,|\,t \in pF\} \\ ZB_{lMax}(p) &= \mathrm{Max}\ \{\ \{ZB_l(p,t)\,|\,t \in pF\}\backslash\{\infty\}\ \} \\ D_{max}(p) &= ZB_{rMax}(p) \Leftrightarrow ZB_{rMax}(p) > ZB_{lMax}(p)\ . \end{aligned} \tag{18}$$

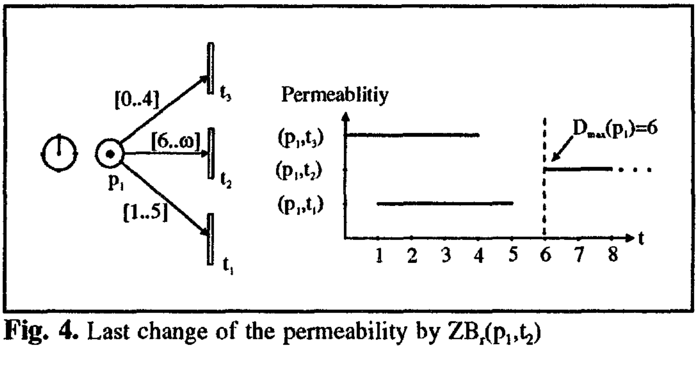

Fig. 4. Last change of the permeability by $ZB_r(p_1,t_2)$

Example

Fig. 4. shows such a case. The last change of the time-dependent permeability occurs when the arc (p_1,t_2) becomes permeable after 6 time units. Any value of $D(p_1)>6$ will not change the permeability of the arcs since the arcs (p_1,t_3) and $p_1,t_1)$ are already unpermeable and the arc (p_1,t_2) stays permeable forever. This situation can only be changed by a change of the marking of place p_1 which resets the clock to zero.

3. The maximal finite time inscription of all arcs (p,t) denotes the *end* of the permeability of an arc. Then $D_{max}(p)$ is equal to the maximal finite value plus one:

$$D_{max}(p) = ZB_{lMax}(p)+1 \Leftrightarrow ZB_{lMax}(p) \geq ZB_{rMax}(p) \ . \tag{19}$$

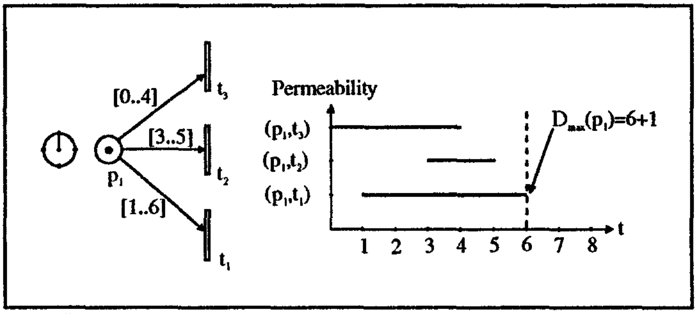

Fig. 5. Last change of the permeability by $ZB_l(p_1,t_1)$

Example

Fig. 5. shows a case where the maximal finite time inscription is given by $ZB_l(p_1,t_1)$. However, the arc is still permeable after 6 time units and becomes unpermeable one time unit later. After seven time units, all arcs are unpermeable. This situation can only be changed by a change of the marking of place p_1 which resets the clock to zero.

The method has some advantages over other timing concepts and analysis methods [9,12]:

1. A new state is generated only by firing of transitions. This reduces the number of states and state transitions compared with methods that generate a new state by any tick of a clock.
2. The concept does not require that all of the arcs $(P \times T) \cap F$ *must* be timed. We can also analyse models which include undelayed firing sequences. Such steps fire "at the same time" but in a causal order.
3. By the modification of the firing rule (16) we define an equivalence of all states with the same markings and clock positions greater than D_{max}. This additionally reduces the number of states and state transitions that must be computed and does not cause a loss of information. For a given initial state and a sequence of state transitions, we can always compute the real clock positions (even if they are greater than D_{max}) by applying equation (12) instead of (16) to compute the resulting clock positions and thereby "unfolding" the equivalent states. This can be performed without additional knowledge of the corresponding Petri net.

Any formal property which is known from the theory of Petri nets can be tested for a timed Petri net by means of the corresponding dynamic graph, but this is beyond the scope of this paper. In the following section, dynamic graphs will only be applied in the sense of control of the modeled system.

4 Performance Analysis and Optimization by Means of Dynamic Graphs

Obviously, any possible firing sequence of state transitions in the dynamic graph represents a discrete process. However, especially for production systems, the state transitions should be performed cyclically and therefore the cyclic stationary behaviour of a system is of special interest. As a basis for the analysis and evaluation of the cyclic stationary behaviour, all cycles c of a dynamic graph must be determined. A cycle c is an ordered set of states and state transitions which contains each state only once:

$$c = ((z_1,g_1),\ldots,(z_i,g_i),\ldots,(z_k,g_k)\,|\,z_i[g_i>z_{i+1}\wedge z_k[g_k>z_1\wedge\forall i,j\in\{1..k\}\ (i\neq j\rightarrow z_i\neq z_j)). \qquad \mathbf{(20)}$$

The word $q(c) = (g_1,\ldots,g_i,\ldots,g_k)$ of the cycle c is a step-invariant of the Petri net:

$$\sum_{g\in q(c)} \Delta u^*(g) = \underline{0}\ . \qquad \mathbf{(21)}$$

The corresponding sum of all step delays $\Delta\tau$ is not zero but denotes the cycle time t_{Cycl} of cycle c:

$$t_{Cycl}(c) = \sum_{g\in q(c)} \Delta\tau(g)\ . \qquad \mathbf{(22)}$$

If a function B defines a value for the contribution of each transition $t \in T$ to an objective function (for instance the output of product):

$$B\colon T \rightarrow \mathbf{N} \qquad \mathbf{(23)}$$

the throughput of the system can be computed for each cycle:

$$d(c) = \frac{\sum_{t\in T} B(t)\ occ(t,c)}{t_{Cycl}(c)} \qquad \mathbf{(24)}$$

where occ(t,c) denotes how often t fires in cycle c.
The cyclic stationary behaviour of the system $\bar{q} = q(c)^\infty$ is the infinite iteration of the word q(c).
Of course, different cycles can be combined to another cyclic stationary behaviour (see Section 5.1).

Modeling of Control Decisions

The dynamic graph of a persistent Petri net model at most contains one cycle. Such a system cannot be influenced by an external control since all conflicts are resolved internally. Since the dynamic graph analysis is applied to compute control strategies, the models which describe the behaviour of the uncontrolled batch process and which are analysed must contain conflicts. Such conflicts represent decisions of a control (see Section 5). Let $T_s \subseteq T$ be the set of controlled transitions. To be controllable, each of this transitions must be in conflict with another controlled transition in at least one state:

$$\forall t \in T_s(\exists t' \in T_s, \exists z \in Z(t' \in cfl(t,z))) \tag{25}$$

where cfl(t,z) denotes the set of transitions which are in conflict with t in state z.
A *control strategy* q_s is a word which only contains controlled transitions and therefore describes a sequence of conflict resolutions (control decisions).
If $T_s \subseteq T$ is the set of controllable transitions, the control strategy $q_s(q,T_s)$ for any process $q = (g_1,\ldots,g_i,\ldots,g_k)$ can be determined:

$$\begin{aligned} u^*(g_{is}) &:= u^*(g_i) \cap T_s \\ \Delta\tau(g_{is}) &= \Delta\tau(g_i). \end{aligned} \tag{26}$$

Modelling of Control Restrictions

Existing *restrictions* for the control strategy can be expressed by *facts*. Facts are transitions which never may be enabled. They have a special graphical symbol (see Section 5). Since facts must be dead, a control strategy must never reach states where facts are enabled. Hence, if the dynamic graph is computed to find out a suitable control strategy, no following states need to be computed when a state is reached where a fact is enabled. The results are smaller dynamic graphs and less effort for the computation of the cycles.

5 Application to Batch Process Control

The application of the above method is based on Petri net models of the batch production systems which must be developed by the chemical engineer, since the chemical engineer has the deepest knowledge of the process which is to be controlled. We will not discuss here how such models can be developed in a systematic way but we will concentrate on the benefit of dynamic graph analysis to determine suitable control strategies.

5.1 Control of a Multiproduct Plant

Fig. 6. shows the flowsheet of a part of a multiproduct batch plant. Product 1 is produced in reactor R1 and product 2 is produced in reactor R2. Both reactors use the metering tanks A and B which contain raw material A and B, respectively. The metering tank C for raw material C is only used by reactor R2.

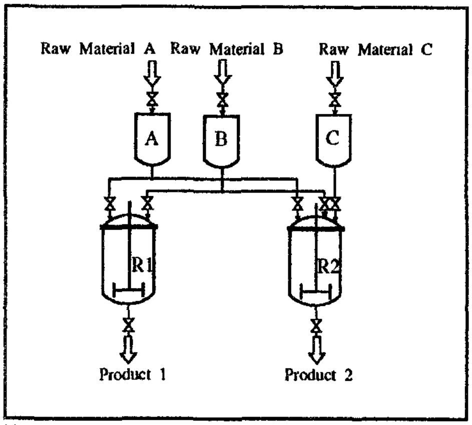

Fig. 6. Flowsheet

The production of product 1 is performed as follows:
At first, raw material A is transfered from metering tank A into reactor R1 (duration: one time unit). After this, raw material B from metering tank B is added (duration: one time unit). Then the reaction is started. When the reaction is completed, the product is discharged (duration: 4 time units), and the cycle starts anew.
For the production of product 2, at first raw material from metering tank C is transfered into reactor R2 (duration: one time unit), then raw material from metering tank B and then from metering tank A is added (duration: each one time unit).
Then the reaction is performed, the product is discharged, and the cycle is started anew (duration: 4 time units).
The metering tanks A and B are shared resources and can be used either by reactor R1 or by reactor R2 but never by both reactors simultaneously.
Fig. 7. shows the Petri net model of the production system. The interpretation of the model is given in Table 1.. Places p_{13}, p_{23} and p_{25} describe that a production process is waiting for an available metering tank. These places must be included into the model to avoid deadlocks.
The initial state is given by the initial marking in Fig. 7.. Since all arcs connecting initially marked places with transitions have the time interval $[0..\infty]$, the local clocks of the marked places are always zero and the clocks need not to be graphically re-

presented.

The model contains structural conflicts between the transitions

- t_{11} and t_{25} and
- t_{13} and t_{23}.

These conflicts must be resolved by a control strategy which must be determined.

Fig. 8. shows a part of the dynamic graph of the net model of Fig. 7.. The complete dynamic graph consists of 47 nodes. The states are not given in detail since they are not necessary for the solution of the control problem. The dotted arcs and the inscriptions $q_{d1}(16)$ and $q_{d2}(6)$ represent sequences of states and state transitions without conflicts. The numbers in parentheses give the times of these two sequences. We can see that the dynamic graph has three cycles (see also Table 2.). These cycles describe elementary stationary control strategies.

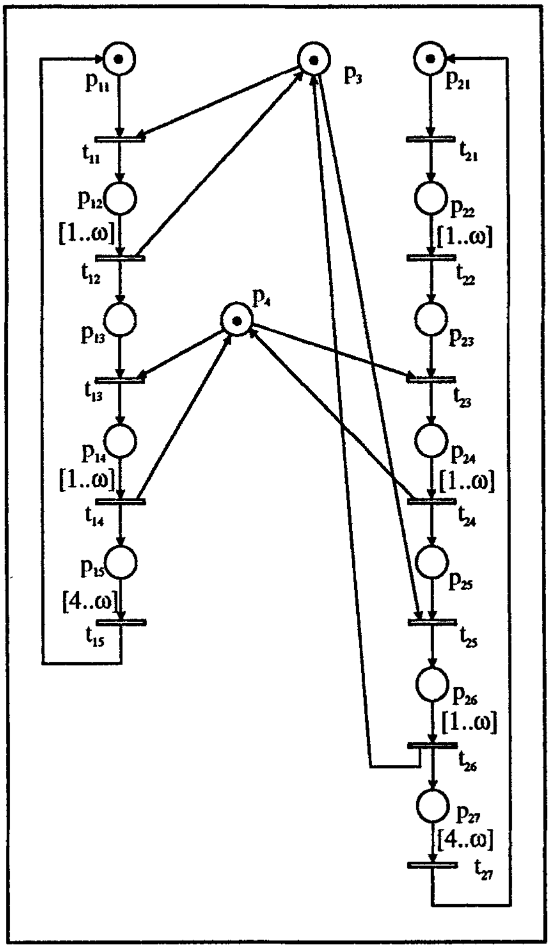

Fig. 7. Petri net model of the production system of Fig. 6.

Transition t_{15} represents the completion of one batch of product 1 whereas transition t_{27} models the completion of one batch of product 2.

Hence, $B(t_{15}) = B(t_{27}) = 1$, $B(t_i) = 0$ for all other transitions. We can compute the throughputs of product 1 and 2, respectively, for each cycle (see Table 2.). The corresponding firing sequences for each cycle can be seen from the dynamic graph (Fig. 8.):

$q(c_1) = (\{t_{11},t_{21}\}(0), \{t_{12},t_{22}\}(1), \{t_{23}\}(0), \{t_{24}\}(1), \{t_{13},t_{25}\}(0), \{t_{14},t_{26}\}(1), \{t_{15},t_{27}\}(4))$

$q(c_2) = (q_{d2}(6), \{t_{25}\}(0), \{t_{26}\}(1))$

$q(c_3) = (\{t_{11},t_{21}\}(0), \{t_{12},t_{22}\}(1), \{t_{13}\}(0), \{t_{14}\}(1), q_{d1}(16), q_{d2}(6), \{t_{11}\}(0), \{t_{12}\}(1), \{t_{13},t_{25}\}(0), \{t_{14},t_{26}\}(1), \{t_{15},t_{27}\}(4))$.

The strategies of cycles c_1 and c_2 resolve the conflicts between t_{11} and t_{25} and between t_{13} and t_{23} in favour of production process 2, cycle c_3 gives priority to production process 1. Consequently, the cycles c_1 and c_2 maximize the throughput of product 2 and the cycle c_3 maximizes the throughput of product 1 (see Table 2.). By combining the different cycles, any throughput of one product between the

p_3	Metering tank A is available	t_{11}	Start of transfer of A into R1
p_4	Metering tank B is available	t_{12}	End of transfer of A into R1
p_{11}	R1 is empty	t_{13}	Start of transfer of B into R1
p_{12}	Transfer of A into R1		
p_{13}	Waiting	t_{14}	End of transfer of B into R1
p_{14}	Transfer of B into R1		
p_{15}	Reaction in R1, Discharge of product	t_{15}	End of product discharge of R1
p_{21}	R2 is empty	t_{21}	Start of transfer of C into R2
p_{22}	Transfer of C into R2		
p_{23}	Waiting	t_{22}	End of transfer of C into R2
p_{24}	Transfer of B into R2		
p_{25}	Waiting	t_{23}	Start of transfer of B into R2
p_{26}	Transfer of A into R2		
p_{27}	Reaction in R2, Discharge of product	t_{24}	End of transfer of B into R2
		t_{25}	Start of transfer of A into R2
		t_{26}	End of transfer of A into R2
		t_{27}	End of product discharge of R2

Table 1. Interpretation of the net model of Fig. 7.

maximal and the minimal value from Table 2. can be realized. Finally, we require that a throughput of product 1 of 0,15 batches per time unit must be realized. A corresponding control strategy shall be determined.

We can see from Table 2. that cycle c_3 causes a production of 5 batches of product 1 in 30 time units and that cycles c_1 and c_2 cause a production of one batch in 7 time units. Let m be the number of realizations of cycle c_3 and n be the number of realizations of cycles c_1 or c_2. The ratio of m to n can be computed by means of the following equation:

$$0,15\frac{\text{Batches of Product 1}}{\text{Time Unit}} = \frac{(5m+n)\ \text{Batches of Product 1}}{(30m+7n)\ \text{Time Unit}}\ . \tag{27}$$

The solution is m:n=1:10. The total time for this strategy is 100 time units and the resulting throughput of product 2 is 0,14 batches per time unit.

The according firing sequence $\bar{q}$ begins with state z_0 and is given as follows: $\bar{q} = (q(c_3),q(c_1)^{10})^{\infty}$. The firing sequences $q(c_3)$ and $q(c_1)$ are given above.

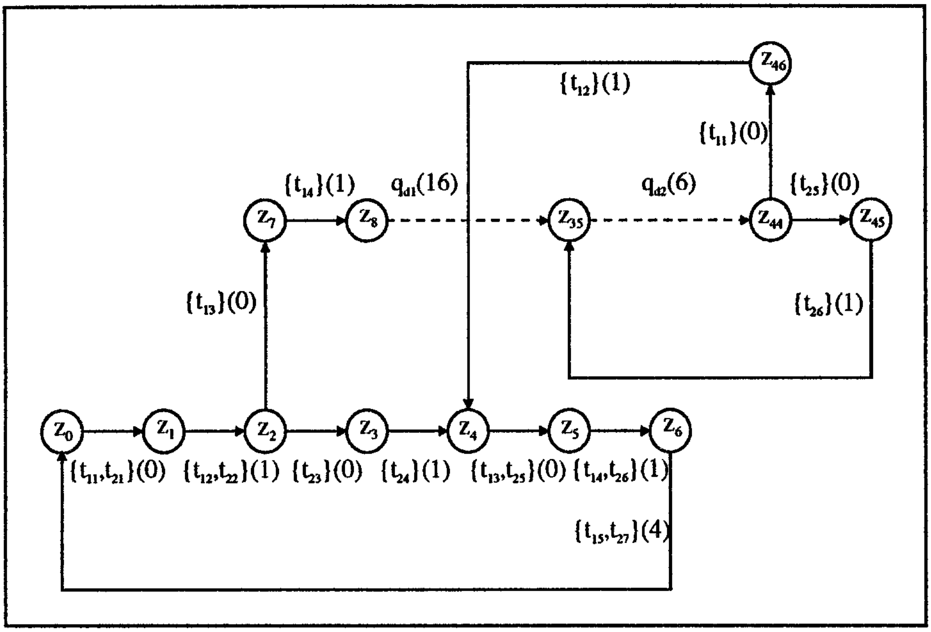

Fig. 8. Dynamic graph of the net model of Fig. 7.

No.	Nodes	t_{Cycl} [Time Units]	Occurrence of		Throughput [Batches/Time Unit]	
			t_{15}	t_{27}	Pro-duct 1	Pro-duct 2
1	$z_0..z_6$	7	1	1	0,143	0,143
2	$z_{35}..z_{45}$	7	1	1	0,143	0,143
3	$z_0..z_2$, $z_7..z_{44}$, z_{46}, z_4 .. z_6	30	5	4	0,167	0,133

Table 2. Comparison of the cycles of the dynamic graph of Fig. 8.

The set of controllable transitions is $T_s = \{t_{11}, t_{13}, t_{23}, t_{25}\}$.
Thus, the corresponding control strategy is (see (26)):
$\bar{q}_s(q,T_s) = (\{t_{11}\}(0), \{\}(1), \{t_{13}\}(0), \{\}(23), \{t_{11}\}(0), \{\}(1), \{t_{13},t_{25}\}(0), \{\}(5), (\{t_{11},t_{21}\}(0), \{\}(1), \{t_{23}\}(0), \{\}(1), \{t_{13},t_{25}\}(0), \{\}(5))^{10})^{\infty}$.
The empty steps reflect that these steps do not require control operations.

5.2 Optimization of Start Times

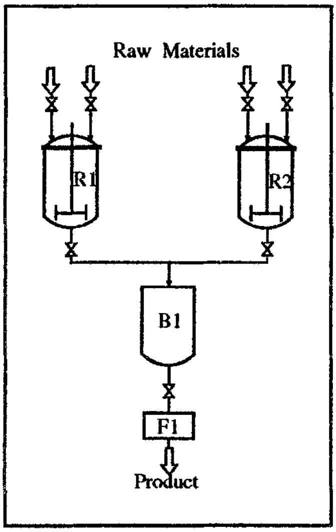

Fig. 9. Flowsheet

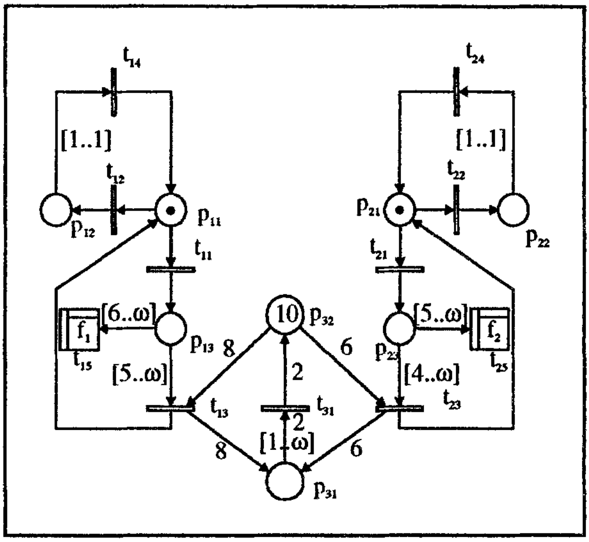

Fig. 10. Petri net model of the production system of Fig. 9.

Fig. 9. shows a plant for the production of a single product. It is produced in two batch reactors (R1 and R2) simultaneously. After completion of the reaction, the product is discharged from the reactors into a storage tank (B1) with a maximal storage capacity of 10 mass units. The product is separated continuously by means of a filter (F1) with a throughput of two mass units per time unit.

p_{11}	Reactor 1 is ready	t_{11}	Start of reaction process R1
p_{12}	Reactor 1 waits		
p_{13}	Reaction in reactor 1	t_{12}	Delay of reaction process R1
p_{21}	Reactor 2 is ready		
p_{22}	Reactor 2 waits	t_{13}	End of reaction process R1
p_{23}	Reaction in reactor 2	t_{21}	Delay of reaction process R2
p_{31}	Storage tank contains $m(p_{31})$ mass units of product	t_{22}	Waiting of reaction process R2
p_{32}	Storage tank has free storage capacity for $m(p_{32})$ mass units of product	t_{23}	End of reaction process R2
		t_{31}	Two mass units of product are separated

Table 3. Interpretation of the net model of Fig. 10.

The reaction processes in the two reactors have different durations (R1: 5 time units, R2: 4 time units) and produce different amounts of product (R1: 8 mass units, R2: 6 mass units). The control has to ensure that the product in reactor R1 or R2, respectively, is discharged immediately after the completion of the reaction. Otherwise, the batch is unusable.

Fig. 10. shows the corresponding Petri net model. The interpretation is given in Table 3.. Control decisions consist of starting the reaction processes (t_{11}, t_{21}) or delaying the start by one time unit (t_{12}, t_{22}) and then decide anew on start or further delay. The technological restrictions described above are modeled by means of facts f_1 and f_2. A suitable control strategy must ensure that these facts are never enabled.

The corresponding dynamic graph was computed by means of ATNA [13]. The result of an analysis of the dynamic graph is a word which describes a stationary behaviour of the system and which is permissible (both facts are never enabled) and optimal (t_{31} fires as often as possible). Fig. 11. shows such a behaviour. The corresponding word q(c) is:

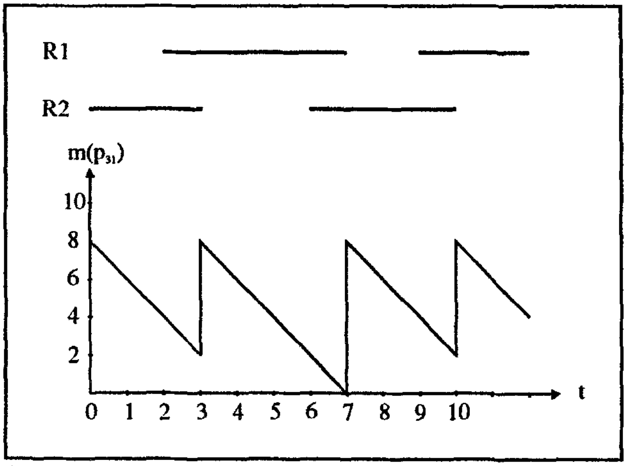

Fig. 11. Control strategy

$q(c) = ((\{t_{12}\}(0), \{t_{14},t_{31}\}(1))^2, \{t_{11}\}(0), \{t_{23},t_{31}\}(1), (\{t_{22}\}(0), \{t_{24},t_{31}\}(1))^3, \{t_{21}\}(0), \{t_{13},t_{31}\}(1))$.

The set of controlled Transitions is $T_s = \{t_{11}, t_{12}, t_{21}, t_{22}\}$.

The control strategy can be computed by means of (26):

$\bar{q}_s(q,T_s) = ((\{t_{12}\}(0), \{\}(1))^2, \{t_{11}\}(0), \{\}(1), (\{t_{22}\}(0), \{\}(1))^3, \{t_{21}\}(0), \{\}(1))^\infty$.

However, the initial state which is given in Fig. 10. is not reproduced by the stationary control strategy. Therefore, a control strategy for the startup phase must be found which ensures that the production is started as soon as possible. A suitable strategy (which was computed by means of ATNA) is to start with reaction process R2 since it takes less time than reaction process R1. This is realized by the word $q(p) = (\{t_{12},t_{21}\}(0), \{t_{14}\}(1), \{t_{12}\}(0), \{t_{14}\}(1), \{t_{11}\}(0), \{t_{23}\}(2), (\{t_{22}\}(0), \{t_{24},t_{31}\}(1))^2, \{t_{21}\}(0), \{t_{13},t_{31}\}(1))$. The duration of the startup strategy is 7 time units. The first product, however, is separated (t_{31} fires) after 5 time units.

The corresponding control strategy is not cyclic but time-optimal for the startup-phase:

$q_s(q,T_s) = (\{t_{12},t_{21}\}(0), \{\}(1), \{t_{12}\}(0), \{\}(1), \{t_{11}\}(0), \{\}(2), (\{t_{22}\}(0), \{\}(1))^2, \{t_{21}\}(0), \{\}(1))$.

One can simulate the token flow in the net of Fig. 10. to prove these results.

6. Conclusions

The result of the analysis of a batch production system by means of dynamic graphs is a suitable or optimal control strategy, given as a time-sequence of firings of controllable transitions. The strategy can be realized by a human operator or an automatic controller. In the second case, the controller must be designed based on the process model (the Petri net) and the control strategy which must be performed. Although some examples for the design of such controllers are given in [3], further work to find a more systematic approach to controller design is necessary.
Although the goal of design is *not* to find a maximal permissive control but to realize an exactly given control strategy, there are close relations to methods proposed in [7,10], and future work will try to apply some of these results to the problem of feedback controller design for systems which are modeled by means of arc-timed Petri nets.
The analysis and optimization method is not restricted to batch process control. The method can also be applied to problems of the manufacturing industry or of computer science which can be modeled by means of arc-timed Place/Transition nets. The computational effort, however, grows exponentially with the size of the model. Thus, the method can only be applied to systems with a small number of control decisions or with a sufficient number of restrictions which can be modeled by facts. A study on the influence of facts on the complexity (for the system of Section 5.2) is given in [4]. It can be seen that the complexity can be reduced drastically by facts, which represent technological knowledge of the process.
Another direction for further research is the problem of online control for disturbance rejection. The basic algorithms of ATNA [13] will be implemented on a process control system for a pilot batch plant to realize a decision support system for the operator. The basic idea is to fire the transitions in the model when the corresponding events in the real process occur. So the marking always represents the current state of the process. If a disturbance occurs, the behaviour of the system can be predicted and appropriate control operations can be performed to prevent dangerous states *before* the system reaches dangerous states. If the time horizon is not too tight, also control strategies for the minimization of losses caused by the disturbance can be computed.

Acknowledgement

The work is supported by the Deutsche Forschungsgemeinschaft under grant Ha 1886/1-1.
The author wishes to thank Prof. P. Starke for the cooperation in the development of ATNA.

References

[1] Hanisch, H.-M.: Dynamik von Koordinierungssteuerungen in diskontinuierlichen verfahrenstechnischen Systemen. at-Automatisierungstechnik 38 (1990), 399-405.

[2] Hanisch, H.-M.: Coordination control modelling in batch production systems by means of Petri nets. Computers and Chemical Engineering, 16 (1992), 1-10.

[3] Hanisch, H.-M.: Petri-Netze in der Verfahrenstechnik. Modellierung und Steuerung verfahrenstechnischer Systeme. R. Oldenbourg Verlag, 1992.

[4] Hanisch, H.-M.: Berechnung optimaler diskreter Koordinierungssteuerungen auf der Grundlage zeitbewerteter Petri-Netze. at-Automatisierungstechnik 40 (1992), 384-390.

[5] Helms, A.; Hanisch, H.-M.: Darstellung einer diskreten Steuerungsaufgabe für flexible Mehrproduktenanlagen mit Hilfe von Petri-Netzen. Chemische Technik 37 (1985), 236-239.

[6] Helms, A.; Hanisch, H.-M.; Stephan, K.: Steuerung von Chargenprozessen. Verlag Technik, Berlin, 1989.

[7] Holloway, L.E.; Krogh, B.H.: Synthesis of feedback control logic for a class of controlled Petri nets. IEEE Transactions on automatic control, Vol. 35, No. 5, May 1990, 514-523.

[8] Le Bail, J.; Alla, H.; David, R.: Hybrid Petri Nets. Proceedings of the European Control Conference, Hermes, Paris, 1991, 1472-1477.

[9] Quäck, L.: Aspekte der Modellierung und Realisierung der Steuerung technologischer Prozesse mit Petri-Netzen. at-Automatisierungstechnik 39 (1991), 116-120, 158-164.

[10] Ramadge, P.J.; Wonham, W.M.: The Control of Discrete Event Systems. Proceedings of the IEEE, Vol. 77, No.1, 1989, pp. 81-98.

[11] Ramchandani, C.: Analysis of asynchronuos concurrent systems by timed Petri nets. MIT, Project MAC, Technical Report 120, 1974.

[12] Starke, P.: Analyse von Petri-Netz-Modellen. B.G. Teubner, Stuttgart, 1990.

[13] Starke, P.: ATNA-Arc Timed Net Analyser. Petri Net Newsletter 37, Dezember 1990, 27-33.

[14] Thiemicke, K.; Hanisch, H.-M.: Prozeßanalyse einer diskontinuierlichen Anlage zur Kunstharzproduktion mit Petri-Netzen. msr, Berlin, 34 (1991), 416-419.

[15] Yamalidou, E.C.; Patsidou, E.P. and Kantor, J.: Modelling discrete-event systems for chemical process control - a survey of several new techniques. Computers and Chemical Engineering 14 (1990), 281-299.

[16] Yamalidou, E.C.; Kantor, J.: Modelling and optimal control of discrete-event chemical processes using Petri nets. Computers and Chemical Engineering 15 (1990), 503-519.