Matching Observed Behavior and Modeled Behavior: An Approach Based on Petri nets and Integer Programming

Wil M.P. van der Aalst

Department of Technology Management
Eindhoven University of Technology
P.O.Box 513, NL-5600 MB Eindhoven, The Netherlands
w.m.p.v.d.aalst@tm.tue.nl

Abstract. Inspired by the way SAP R/3 and other transactional information systems log events, we focus on the problem to decide whether a process model and a frequency profile "fit" together. The problem is formulated in terms of Petri nets and an approach based on integer programming is proposed to tackle the problem. The integer program provides necessary conditions and, as shown in this paper, for relevant subclasses these conditions are sufficient. Unlike traditional approaches, the approach allows for labelled Petri nets with "hidden transitions", noise, etc.

Keywords: Concurrency, Distributed systems, Petri nets, Integer programming, ERP, Marking equation.

1 Introduction

For many processes in practice there exist models. These model are *descriptive* or *prescriptive*, i.e., they are used to describe a process or they are used to control or guide the system. A typical example are the so-called reference models in the context of Enterprise Resource Planning (ERP) systems like SAP [6]. The SAP reference models are expressed in terms of so-called Event-driven Process Chains (EPCs) describing how people should/could use the SAP R/3 system. Similarly models are used in the workflow domain [1], but also in many other domains ranging from flexible manufacturing and telecommunication to operating systems and software components [7]. In some domains these models are referred to as *specifications* or *blueprints*. In reality, the real process may deviate from the modeled process, e.g., the implementation is not consistent with the specification or people use SAP R/3 in a way not modeled in any of the EPCs.

Clearly, the problem of checking whether the modeled behavior and the observed behavior match is not new. However, when we applied our process mining techniques [2] to SAP R/3 we where confronted with the following interesting problem: The logs of SAP do not allow for monitoring individual cases (e.g.,

purchase orders). Instead SAP only logs the fact that a specific transaction has been executed (without referring to the corresponding case). Hence, tools like the SAP Reverse Business Engineer (RBE) report on the frequencies of transaction types and not on the cases themselves. These transactions can be linked to functions in the EPCs, but, as indicated, not to individual cases. Moreover, some functions in the EPC do not correspond to a transaction code, and therefore, are not logged at all. This raises the following interesting question: *Do the modeled behavior (i.e., the EPC) and the observed behavior (i.e., the transaction frequencies) match?*

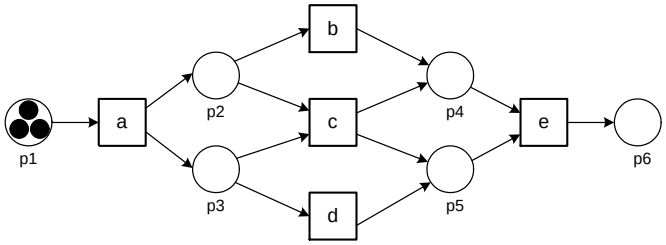


Fig. 1. A Petri net.

In this paper we consider an abstraction of the problem. Consider a *Petri net* with some initial marking [8, 9] and a *frequency profile* which is a partial function indicating how many times certain transitions fired. Consider for example the marked Petri net shown Figure 1. A frequency profile fp could be $fp(a) = 3$, $fp(b) = 2$, $fp(c) = 2$, $fp(d) = 2$, and $fp(e) = 3$, thus indicating the number of times each transition occurred. However, the modeled behavior (i.e., the marked Petri net) and the observed behavior (the frequency profile fp) do not match. It is easy to see that $fp(b) + fp(c)$ cannot exceed $fp(a)$ since b and c depend on the tokens produced by a. Now consider another frequency profile fp: $fp(a) = 3$, $fp(b) = 2$, $fp(d) = 2$, and $fp(e) = 3$, i.e., the number of times c occurred is unknown. Now the modeled behavior and the observed behavior match, i.e., the observed transition frequencies are consistent with the Petri net model. Moreover, it is clear that in this situation c occurred precisely once.

In the remainder we will focus on this problem and propose an approach based on *Integer Programming* (IP) [11, 13]. Using a marked Petri net and a frequency profile, an IP problem is formulated to check whether the modeled behavior and the observed behavior match and, if so, the frequency of transitions not recorded in the profile is determined. First, we introduce some basic Petri net notations. Then, we formulate the IP problem and demonstrate its applicability using an example. Finally, we briefly discuss related work and provide some final remarks on practical relevance of the results.

2 Petri nets

This section introduces the basic Petri net terminology and notations (cf. [9, 4]). Readers familiar with Petri nets can skip this section.

The classical Petri net is a directed bipartite graph with two node types called *places* and *transitions*. The nodes are connected via directed *arcs*. Connections between two nodes of the same type are not allowed. Places are represented by circles and transitions by rectangles.

Definition 1 (Petri net). *A Petri net is a triple* (P, T, F)*:*

- P *is a finite set of places,*
- T *is a finite set of transitions* ($P \cap T = \emptyset$)*,*
- $F \subseteq (P \times T) \cup (T \times P)$ *is a set of arcs (flow relation)*

A place p is called an *input place* of a transition t iff there exists a directed arc from p to t. Place p is called an *output place* of transition t iff there exists a directed arc from t to p. We use $\bullet t$ to denote the set of input places for a transition t. The notations $t\bullet$, $\bullet p$ and $p\bullet$ have similar meanings, e.g., $p\bullet$ is the set of transitions sharing p as an input place. In this paper, we do not consider multiple arcs from one node to another. However, all results can be extended to Petri nets with arcs weights.

Figure 1 shows a Petri net with 5 transitions (a, b, c, d, and e) and 6 places ($p1, \ldots p6$).

At any time a place contains zero or more *tokens*, drawn as black dots. The *state*, often referred to as *marking*, is the distribution of tokens over places, i.e., $M \in P \to \mathbb{N}$. We will represent a marking as follows: $1'p1 + 2'p2 + 1'p3 + 0'p4$ is the marking with one token in place $p1$, two tokens in $p2$, one token in $p3$ and no tokens in $p4$. We can also represent this marking as follows: $p1 + 2'p2 + p3$. The marking shown in Figure 1 is $p1$. (Note the overloading of notation.) To compare markings we define a partial ordering. For any two markings M_1 and M_2, $M_1 \leq M_2$ iff for all $p \in P$: $M_1(p) \leq M_2(p)$.

The number of tokens may change during the execution of the net. Transitions are the active components in a Petri net: they change the marking of the net according to the following *firing rule*:

(1) A transition t is said to be *enabled* iff each input place p of t contains at least one token.
(2) An enabled transition may *fire*. If transition t fires, then t *consumes* one token from each input place p of t and *produces* one token for each output place p of t.

In Figure 1 transition a is enabled. Firing a results in marking $2'p1 + p2 + p3$. In this marking, three additional transitions (besides a) are enabled (b, c, d). Any of these transitions may fire. However, firing one of these transition will disable one or two other transitions, e.g., firing c will disable both b and d.

Given a Petri net (P, T, F) and a marking M_1, we have the following notations:

- $M_1 \xrightarrow{t} M_2$: transition t is enabled in marking M_1 and firing t in M_1 results in marking M_2
- $M_1 \rightarrow M_2$: there is a transition t such that $M_1 \xrightarrow{t} M_2$
- $M_1 \xrightarrow{\sigma} M_n$: the firing sequence $\sigma = t_1 t_2 t_3 \ldots t_{n-1}$ leads from marking M_1 to marking M_n via a (possibly empty) set of intermediate markings $M_2, \ldots M_{n-1}$, i.e., $M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} \ldots \xrightarrow{t_{n-1}} M_n$

A marking M_n is called *reachable* from M_1 (notation $M_1 \xrightarrow{*} M_n$) iff there is a firing sequence σ such that $M_1 \xrightarrow{\sigma} M_n$. Note that the empty firing sequence is also allowed, i.e., $M_1 \xrightarrow{*} M_1$.

To manipulate firing sequences, we introduce the *Parikh vector* $\pi_\sigma \in T \rightarrow \mathbb{N}$, where $\pi_\sigma(t)$ denotes the number of occurrences of transition t in σ.

We use (PN, M) to denote a Petri net PN with an initial marking M. A marking M' is a *reachable marking* of (PN, M) iff $M \xrightarrow{*} M'$. Consider the Petri net shown in Figure 1 with only one token in $p1$. For this initial marking there are 6 reachable markings.

3 Matching a marked Petri net and a frequency profile

Petri nets may be used to model a wide variety of processes. A Petri net can model what we think the process is (i.e., descriptive) but it can also model what the process should be (i.e., prescriptive). In both cases, the real process may deviate from what is modeled in the Petri net. In this section, we investigate whether the modeled behavior (i.e., Petri net) and the observed behavior match. Since in reality we often cannot inspect the state and just observe events, it is realistic to assume that we can only monitor the firing of transitions. Moreover, we assume that we cannot link transition occurrences to specific tokens or exploit their ordering in time, i.e., we only know the *frequency profile*. For a Petri net with transitions T the frequency profile refers to a subset of T, i.e., frequency profile $fp \in T \not\rightarrow \mathbb{N}$ is a partial function. For $t \in dom(fp)$, $fp(t)$ is the number of times t occurred/fired. For $t \notin dom(fp)$ this is unknown. If $dom(fp) = T$, the frequency profile is *complete*. Both for complete and incomplete frequency profiles we define the predicate $match(PN, M, fp)$.

Definition 2 (Match). *Let (PN, M) be a marked Petri net with $PN = (P, T, F)$ and $fp \in T \not\rightarrow \mathbb{N}$ a frequency profile. (PN, M) and fp match if there exists a firing sequence σ enabled in M (i.e., $M \xrightarrow{\sigma}$) such that for all $t \in dom(fp)$: $fp(t) = \pi_\sigma(t)$. (Notation: $match(PN, M, fp)$.)*

In the introduction we mentioned two frequency profiles for the marked Petri net shown in Figure 1. The first one (i.e., $fp(a) = 3$, $fp(b) = 2$, $fp(c) = 2$, $fp(d) = 2$, and $fp(e) = 3$) does not match while the second one (i.e., $fp(a) = 3$, $fp(b) = 2$, $fp(d) = 2$, and $fp(e) = 3$) does. Note that the first profile is complete while the second is incomplete ($c \notin dom(fp)$). For any marked Petri net there is a trivial matching profile fp with $dom(fp) = \emptyset$.

Even for moderate examples, the number of firing sequences may be too large to check $match(PN, M, fp)$. Therefore, in the spirit of [3, 7], we can try to formulate a linear algebraic representation. Given the discrete nature of firing transitions, we propose an Integer Programming (IP) problem rather than an Linear Programming (LP) problem [11, 13].

Definition 3 (Integer programming problem). *Let (PN, M) be a marked Petri net with $PN = (P, T, F)$ and $fp \in T \not\to \mathbb{N}$ a frequency profile. $IP(PN, M, fp)$ is the corresponding Integer Programming (IP) problem:*

$$
\begin{array}{ll}
\min \sum_{t \in T} f_t & \\
\text{s.t. } f_t = fp(t) & \text{for all } t \in dom(fp) \\
\quad f_{(t,p)} = f_t & \text{for all } (t,p) \in F \cap (T \times P) \\
\quad f_{(p,t)} = f_t & \text{for all } (p,t) \in F \cap (P \times T) \\
\quad M(p) + \sum_{t \in {}^\bullet p} f_{(t,p)} - \sum_{t \in p^\bullet} f_{(p,t)} \geq 0 & \text{for all } p \in P \\
\quad f_t \geq 0 & \text{for all } t \in T \\
\quad f_t & \text{integer for all } t \in T \\
\quad f_{(x,y)} & \text{integer for all } (x,y) \in F
\end{array}
$$

There are two types of positive integer variables: f_t for transition frequencies and $f_{(x,y)}$ for arc frequencies. The first constraint specifies that the transition frequencies should match the frequency profile. Note that for some transitions there may not be a frequency in the frequency profile. The second and third constraint refer to the fact that transition frequencies and arc frequencies need to be aligned. The fourth type of constraint is the most interesting one. For each place, there should be a balance between the inflow of tokens and the outflow of tokens, i.e., it is not possible to consume more tokens than the initial ones plus the produced ones. The objective function minimizes the number of firings. Given the nature of the problem this is of less importance and alternative objective functions can be defined, e.g., an objective function maximizing or minimizing the number of tokens in the net.

Before we discuss the relation between $match(PN, M, fp)$ and $IP(PN, M, fp)$, let us return to the Petri net shown in Figure 1. Assuming some initial marking

M and some frequency profile fp, $IP(PN, M, fp)$ is formulated as follows.

$$
\begin{aligned}
\min\ & f_a + f_b + f_c + f_d + f_e \\
\text{s.t. } & f_a = fp(a) \\
& \ldots \\
& f_{(a,p2)} = f_a \\
& \ldots \\
& f_{(p1,a)} = f_a \\
& \ldots \\
& M(p1) - f_{(p1,a)} \geq 0 \\
& M(p2) + f_{(a,p2)} - f_{(p2,b)} - f_{(p2,c)} \geq 0 \\
& M(p3) + f_{(a,p3)} - f_{(p3,c)} - f_{(p3,d)} \geq 0 \\
& M(p4) + f_{(b,p4)} + f_{(c,p4)} - f_{(p4,e)} \geq 0 \\
& M(p5) + f_{(c,p5)} + f_{(d,p5)} - f_{(p5,e)} \geq 0 \\
& M(p6) + f_{(e,p6)} \geq 0 \\
& f_a \geq 0 \\
& \ldots \\
& f_a \quad \text{integer} \\
& \ldots \\
& f_{(p1,a)} \quad \text{integer} \\
& \ldots
\end{aligned}
$$

Applying this to the initial marking shown in Figure 1 and the frequency profile $fp(a) = 3$, $fp(b) = 2$, $fp(c) = 2$, $fp(d) = 2$, and $fp(e) = 3$ indeed results in an IP problem without a solution. While applying it to the second frequency profile $fp(a) = 3$, $fp(b) = 2$, $fp(d) = 2$, and $fp(e) = 3$ yields the solution where $f_c = 1$. In the latter case the value of the objective function is 11.

In the remainder of this section we investigate the relation between $match(PN, M, fp)$ and $IP(PN, M, fp)$, i.e., "Can the IP problem be used to determine whether the modeled and observed behavior match?". The following theorem shows that, as expected, the IP problem indeed provides necessary requirements.

Theorem 1. *Let (PN, M) be a marked Petri net with $PN = (P, T, F)$ and $fp \in T \not\to \mathbb{N}$ a frequency profile. If $match(PN, M, fp)$, then $IP(PN, M, fp)$ has a solution.*

Proof. If $match(PN, M, fp)$, then there exists a firing sequence σ enabled in M (i.e., $M \xrightarrow{\sigma}$) such that for all $t \in T$: $fp(t) = \pi_\sigma(t)$. Let M' be the resulting marking. Now consider the IP problem. The only constraint that could be violated is $M(p) + \sum_{t \in {}^\bullet p} f_{(t,p)} - \sum_{t \in p^\bullet} f_{(p,t)} \geq 0$ for some $p \in P$. However, this constraint follows directly from the firing rule. In fact, $M(p) + \sum_{t \in {}^\bullet p} f_{(t,p)} - \sum_{t \in p^\bullet} f_{(p,t)} = M'(p)$. □

The result does not hold in the opposite direction, as can be shown by an example taken from [4]. Figure 2 shows a marked Petri net. Let $fp(t) = 1$ for all transitions t except for $t = g$ which occurs twice (i.e., $fp(g) = 2$). It is easy to verify that $IP(PN, M, fp)$ has a solution. However, the marked Petri net and the

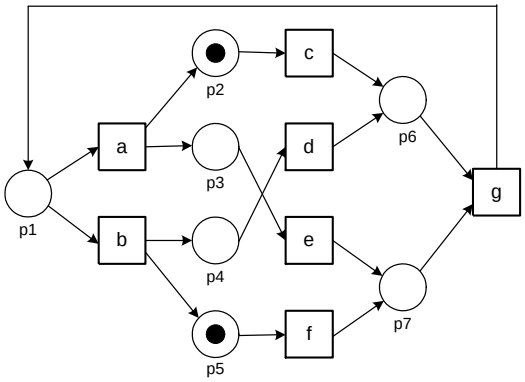


Fig. 2. Counter example.

frequency profile do not match because there is no firing sequence (starting in the initial marking shown in Figure 2) that fires g twice and all other transitions once. (Note that it is impossible to return to the initial marking.) Fortunately, for certain subclasses the result does hold in the opposite direction as is illustrated by the following theorem.

Theorem 2. *Let (PN, M) be an acyclic marked Petri net with $PN = (P, T, F)$ and $fp \in T \not\to \mathbb{N}$ a frequency profile such that $IP(PN, M, fp)$ has a solution. There exists a firing sequence σ enabled in M such that for all $t \in dom(fp)$: $fp(t) = \pi_\sigma(t)$, i.e., $match(PN, M, fp)$.*

Proof. In the solution of $IP(PN, M, fp)$ each transition $t \in T$ fires f_t times. Let $n = \sum_{t \in T} f_t$. If $n = 0$, the empty sequence is enabled and the theorem holds. If $n > 0$, remove all transitions t for which $f_t = 0$. Moreover, remove all places and arcs not connected to a transition t for which $f_t > 0$. Let PN' be the resulting net and M' the resulting marking. Clearly, PN' is acyclic. At least one transition is enabled in (PN', M'). (If not, the fact that PN' is acyclic would imply that there is an empty source place p with some output transition t'. However, $M(p) + \sum_{t \in {}^\bullet p} f_{(t,p)} - \sum_{t \in p^\bullet} f_{(p,t)} = M'(p) + 0 - f_{(p,t')} - \ldots = 0 + 0 - f_{t'} - \ldots \geq 0$. Clearly, this leads to a contradiction.) Fire this enabled transition t^* and let M^* be the resulting marking and fp^* such that $fp^*(t^*) = fp(t^*) - 1$ and for all other $t \in dom(fp)$: $fp^*(t) = fp(t)$. Clearly, $IP(PN, M^*, fp^*)$ has a solution. Repeat the above process until $n = 0$. In each step, a transition t^* is fired thus forming a sequence σ enabled in M. □

Note that the proof of this theorem is similar to Theorem 16 in [7]. Consider Figure 2 with the arc from g to $p1$ removed and a new place $p8$ added as an output place of g. Now for any marking M and any frequency profile fp such that $IP(PN, M, fp)$ has a solution, there exists a corresponding firing sequence, i.e., $match(PN, M, fp)$. For example, given the marking shown in Figure 2 and the acyclic variant of the net, the IP problem has a solution for the following

frequency profile fp: $fp(a) = fp(b) = fp(d) = fp(e) = 0$, $fp(c) = fp(f) = fp(g) = 1$. Indeed, as suggested by Theorem 2, there is a firing sequence firing c, f and g (e.g., cfg).

The counter example shown in Figure 2 is free-choice [4]. Therefore, one could consider to proving Theorem 2 for subclasses of free-choice nets (i.e., replace the requirement that the net is acyclic with some other structural requirement). Two well-known subclasses are the class of *marked graphs* and the class of *state machines* [4, 7, 9].

A *marked graph* is a Petri net with for each place $p \in P$: $|\bullet p| = |p\bullet| = 1$ (i.e., places cannot have multiple input or output transitions). A *circuit* is a circular path in the Petri net such that no element (i.e., place or transition) occurs more than once. It is easy to see that in a marked graph the number of tokens in a circuit is constant. Therefore, a circuit remains (un)marked if it is (un)marked in the initial marking. Using existing results it is easy to prove that Theorem 2 applies to (cyclic) marked graphs where each circuit is marked.

Theorem 3. *Let (PN, M) be an marked graph with $PN = (P, T, F)$ and $fp \in T \not\to \mathbb{N}$ a frequency profile. If each circuit is initially marked, then $IP(PN, M, fp)$ has a solution if and only if $match(PN, M, fp)$.*

Proof. As shown in Theorem 1, $match(PN, M, fp)$ implies that $IP(PN, M, fp)$ has a solution. Remains to prove that $IP(PN, M, fp)$ has a solution also implies $match(PN, M, fp)$. Consider a solution assigning values to each f_t and $f_{(x,y)}$. Let M' be a marking defined as follows: $M(p) + \sum_{t \in \bullet p} f_{(t,p)} - \sum_{t \in p\bullet} f_{(p,t)} = M'(p)$ for all $p \in P$. Note that M' is indeed a marking, i.e., for each $p \in P$, $M'(p)$ is a non-negative integer. This implies that the marking equation $M + N.X = M'$ has a solution. (N is the incidence matrix and X is a vector.) This solution is given by the values assigned to f_t. Because there is a solution, M and M' agree on all place invariants. For live marked graphs a marking M' is reachable from M if and only if both agree on all place invariants (cf. Theorem 3.21 in [4]). A marked graph where each circuit is initially marked is live (cf. Theorem 3.15 in [4]). Therefore, M' is reachable from M and $match(PN, M, fp)$. □

Figure 3 shows a marked graph. For any initial marking M, the IP problem has a solution if and only if $match(PN, M, fp)$ (provided that every circuit is initially marked).

A Petri net is a *state machine* iff transitions cannot have more than one input or output place, i.e., for each transition $t \in T$: $|\bullet t| = |t\bullet| = 1$. It is easy to prove that Theorem 3 also holds for state machines as long as the the net is strongly connected (i.e., there is a directed path from any node to any other node in the net) and initially there is at least one token.

Theorem 4. *Let (PN, M) be a strongly-connected state machine with $PN = (P, T, F)$ and a non-empty initial marking M and $fp \in T \not\to \mathbb{N}$ a frequency profile. $IP(PN, M, fp)$ has a solution if and only if $match(PN, M, fp)$.*

Proof. As shown in Theorem 1, $match(PN, M, fp)$ implies that $IP(PN, M, fp)$ has a solution. Remains to prove that the reverse also holds. Consider a solution

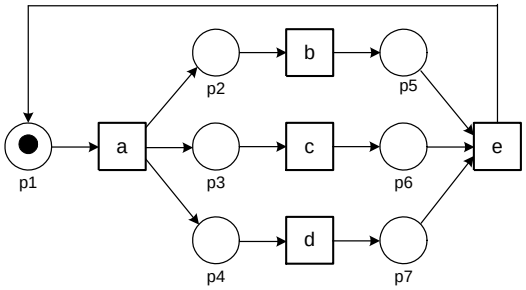


Fig. 3. Marked graph.

assigning values to each f_t and $f_{(x,y)}$. Let M' be a marking defined as follows: $M(p) + \sum_{t \in {}^\bullet p} f_{(t,p)} - \sum_{t \in p^\bullet} f_{(p,t)} = M'(p)$ for all $p \in P$. Note that M' is indeed a marking, i.e., for each $p \in P$, $M'(p)$ is a non-negative integer. The number of tokens in M equals the number of tokens in M', in fact M and M' agree on all place invariants. Moreover, the marked state machine is live because PN is a strongly-connected state machine and M is non-empty (cf. Theorem 3.3 in [4]). Using the second reachability theorem (cf. Theorem 3.8 in [4]), it follows that M' is reachable from M and $match(PN, M, fp)$. □

Figure 4 shows a strongly connected state machine. For any non-empty initial marking M $IP(PN, M, fp)$ has a solution if and only if $match(PN, M, fp)$.

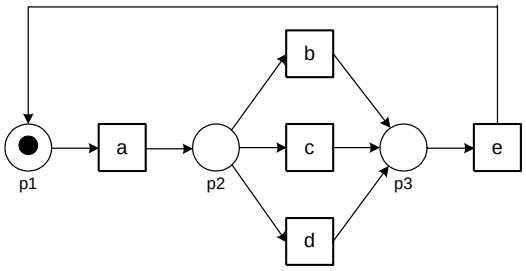
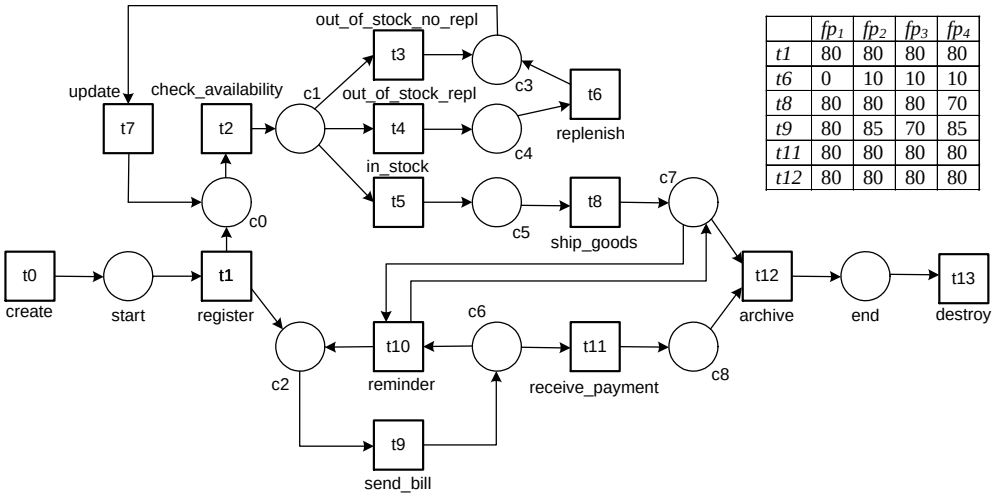


Fig. 4. State machine.

In this section, we explored the relation between $match(PN, M, fp)$ (i.e., the predicate indicating that a process model and observed transition frequencies fit together) and $IP(PN, M, fp)$ (i.e., an integer programming problem). In the remainder, we consider a larger example, possible extensions, and related work.

4 Example

After showing a number of abstract examples, we now use the more realistic example shown in Figure 5. It describes the workflow [1] of handling orders. The upper half models the logistical subprocess while the lower half models the financial subprocess. Most of the workflow should be self explanatory except perhaps for the construct involving $c7$ and $t10$ (*reminder*): A reminder can only be sent if the goods have been shipped.



	fp_1	fp_2	fp_3	fp_4
t1	80	80	80	80
t6	0	10	10	10
t8	80	80	80	70
t9	80	85	70	85
t11	80	80	80	80
t12	80	80	80	80

Fig. 5. A Petri net modeling the processing of customer orders and four frequency profiles.

Unlike the other two Petri nets, the initial marking is empty. Instead a source and a sink transition have been added. Transition $t0$ (*create*) creates the order while $t13$ (*destroy*) marks the end of the order. This pattern is often used to model an unknown number of cases.

Suppose that only the steps $t1$ (*register*), $t6$ (*replenish*), $t8$ (*ship_goods*), $t9$ (*send_bill*), $t11$ (*receive_payment*), and $t12$ (*archive*) are recorded. Figure 5 shows four frequency profiles (fp_1, fp_2, fp_3, and fp_4). The IP problems corresponding to the first two profiles (fp_1 and fp_2), both have a solution. It is also easy to see that fp_1 and fp_2 both indeed match with the Petri net. Note that in the first profile there are no replenishment orders and no reminders, i.e., $t4$, $t6$ and $t10$ do not fire. It is also interesting to note that the number of times $t3$ and $t7$ fire is not constrained by fp_1, however, by the objective function their frequencies are set to 0. In the second profile there are 10 replenishment orders and 5 reminders. The IP problems corresponding to the last two profiles (fp_3 and fp_4), both do not have a solution and, indeed, fp_3 and fp_4 do not match with the Petri net.

In fp_3 there are not enough bills (70) to justify the number of payments (80). In fp_4 there are not enough shipments.

5 Extensions

A Linear Programming (LP) problem can be solved in polynomial time while an IP problem is NP complete [11, 13]. Therefore, it may be interesting to consider the LP relaxation of $IP(PN, M, fp)$. We expect that in some cases this will provide good results. Note that often the rounded LP relaxation provides a feasible but non-optimal solution (but not always, cf. the example net shown on page 269 in [3]). Since the objective function is of less interest, this is not a problem. Also note that if the IP problem has a solution the LP problem will also have a solution. Therefore, Theorem 1 also holds for the LP relaxation. As a result the LP problem can be used to quickly point out discrepancies between the process model and the frequency profile.

The LP relaxation is also interesting if the frequency profile is not exact or if we want to abstract from exceptions, i.e., if we consider *noise* we are not interested in the exact number of firings but in an approximate number. Suppose we want to allow a margin of 10 percent. To specify this we replace the first constraint in Definition 3 ($f_t = fp(t)$) by two weaker constraints: $f_t \geq 0.9fp(t)$ and $f_t \leq 1.1fp(t)$. Such approximations are also needed if we collect data for a limited period with an unknown number of tokens in the initial marking.

Definition 4. *Let (PN, M) be a marked Petri net with $PN = (P, T, F)$, $fp \in T \not\to \mathbb{N}$ a frequency profile, and α the noise level ($0 \leq \alpha \leq 1$) . The corresponding LP (IP) problem allowing for α noise:*

$$
\begin{array}{ll}
\min \sum_{t \in T} f_t & \\
\text{s.t. } f_t \geq (1-\alpha) fp(t) & \text{for all } t \in dom(fp) \\
\quad f_t \leq (1+\alpha) fp(t) & \text{for all } t \in dom(fp) \\
\quad f_{(t,p)} = f_t & \text{for all } (t,p) \in F \cap (T \times P) \\
\quad f_{(p,t)} = f_t & \text{for all } (p,t) \in F \cap (P \times T) \\
\quad M(p) + \sum_{t \in {}^\bullet p} f_{(t,p)} - \sum_{t \in p^\bullet} f_{(p,t)} \geq 0 & \text{for all } p \in P \\
\quad f_t \geq 0 & \text{for all } t \in T \\
\quad f_t & \text{(integer) for all } t \in T \\
\quad f_{(x,y)} & \text{(integer) for all } (x,y) \in F
\end{array}
$$

Note that Definition 4 defines both an LP and and IP problem. The only difference is that for the LP problem the variables do not need to be integers.

Another extension is the situation where multiple transitions refer to the same event, e.g., in SAP multiple functions in the EPC may generate the same transaction. This corresponds to a labeled Petri net with multiple transitions having the same label. Again this is easy to incorporate in the IP problem. The frequency profile is no longer a mapping from transitions to frequencies but from transition labels to frequencies and the first constraint should be replaced as indicated below.

Definition 5. *Let (PN, M) be a marked Petri net with $PN = (P, T, F)$, L a set of labels, $lab \in T \not\to L$ a labeling function, and $fp \in L \not\to \mathbb{N}$ a frequency profile. The corresponding IP problem is:*

$$
\begin{array}{ll}
\min & \sum_{t \in T} f_t \\
\text{s.t.} & \sum_{t \in dom(lab) \mid lab(t) = l} f_t = fp(l) \quad \text{for all } l \in L \\
& f_{(t,p)} = f_t \quad \text{for all } (t,p) \in F \cap (T \times P) \\
& f_{(p,t)} = f_t \quad \text{for all } (p,t) \in F \cap (P \times T) \\
& M(p) + \sum_{t \in {}^\bullet p} f_{(t,p)} - \sum_{t \in p^\bullet} f_{(p,t)} \geq 0 \quad \text{for all } p \in P \\
& f_t \geq 0 \quad \text{for all } t \in T \\
& f_t \quad \text{integer for all } t \in T \\
& f_{(x,y)} \quad \text{integer for all } (x,y) \in F
\end{array}
$$

All results given in Section 3 can be extended to labeled Petri nets.

Note that definitions 4 and 5 can be combined. These extensions show that the formulation in terms of an LP/IP problem is easy to refine or extend.

6 Related work

The work presented is most related to the "Marking Equation" known from Petri net theory [7, 3, 12] and this paper builds on some of these results. However, the approach presented differs in at least two ways. First of all, the marking equation considers the initial *and* resulting marking while we only consider the initial marking. Second, we allow for transition frequencies that are unknown, i.e., the frequency profile may be incomplete. Moreover, the approach allows for the extensions described in Section 5 while the marking equation does not. Clearly there are also relations with the classical results on place and transition invariants [4, 12, 8]. However, these are less direct. As indicated in the introduction, the problem addressed resulted from the application of process mining techniques [2] to SAP. This was done in the context of configurable process models, cf. [10] for more details.

7 Conclusion

Inspired by a problem encountered when applying process mining techniques to SAP transaction logs, the paper tackled the problem of checking whether a Petri net and a frequency profile match. An IP problem was proposed to efficiently implement a necessary but not sufficient condition. The approach allows for extensions not possible in the traditional linear algebraic approaches [7, 3, 12]. Clearly, the application is not limited to SAP transaction logs but is applicable in any situation where processes are only monitored at an aggregate level, i.e., frequency profiles rather than event traces.

Future research is aiming at a better characterization of the class of nets for which $IP(PN, M, fp)$ has a solution if and only if $match(PN, M, fp)$. In this paper, it was shown that for acyclic nets, marked graphs, and state machines

this is the case. It seems that the characterizations given in [5] and the class of ST-nets (nets obtained by composing marked graphs and state machines) are a good starting point for a beter understanding when solutions of the IP problem are actually realizable.

Acknowledgments. The author would like to thank Eric Verbeek of proofreading the paper and Monique Jansen-Vullers and Michael Rosemann for their joint work on mining SAP and configurable process models which uncovered the problem addressed in this paper.

References

1. W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
2. W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003.
3. J. Desel. Basic Linear Algebraic Techniques of Place/Transition Nets. In W. Reisig and G. Rozenberg, editors, *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*, pages 257–308. Springer-Verlag, Berlin, 1998.
4. J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1995.
5. K. van Hee, N. Sidorova, and M. Voorhoeve. Soundness and Separability of Workflow Nets in the Stepwise Refinement Approach. In W.M.P. van der Aalst and E. Best, editors, *Application and Theory of Petri Nets 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 335–354. Springer-Verlag, Berlin, 2003.
6. G. Keller and T. Teufel. *SAP R/3 Process Oriented Implementation*. Addison-Wesley, Reading MA, 1998.
7. T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.
8. W. Reisig. *Petri Nets: An Introduction*, volume 4 of *EATCS Monographs in Theoretical Computer Science*. Springer-Verlag, Berlin, 1985.
9. W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1998.
10. M. Rosemann and W.M.P. van der Aalst. A Configurable Reference Modelling Language. QUT Technical report, FIT-TR-2003-05, Queensland University of Technology, Brisbane, 2003.
11. A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, New York, 1998.
12. M. Silva, E. Teruel, and J.M. Colom. Linear Algebraic and Linear Programming Techniques for the Analysis of Place/Transition Net Systems. In W. Reisig and G. Rozenberg, editors, *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*, pages 309–373. Springer-Verlag, Berlin, 1998.
13. L.A. Wolsey. *Integer Programming*. John Wiley & Sons, New York, 1998.