

Estimating costs of a service

Christian Gierds and Jan Sürmeli

Institut für Informatik
Humboldt-Universität zu Berlin
Unter den Linden 6, 10099 Berlin, Germany
{gierds|suermeli}@informatik.hu-berlin.de

Abstract. When designing a publicly available Web service, a service designer has to take care of costs and revenue caused by this services. In the very beginning possible partners might only be vaguely known, or the service behavior contains arbitrary repetitions. Then the estimation of costs for running this service is difficult and decisions based on them can hardly be made.

We propose a static analysis of the service's behavior. We over-approximate possible runs and therefore costs of the service. Our approach provides a basis for reasoning about nonfunctional properties as shown for costs.

Key words: Nonfunctional properties, state equation, open nets, service behavior

1 Motivation and approach

A *service provider* in a *service-oriented architecture* (SOA) [1] has only limited knowledge about the future interaction partners of its published service. Then determining nonfunctional properties such as estimating the costs of a service run (or any other performance measure) in the interaction with a *service requester* is complex, because they vary depending on the requester's behavior. A *service broker* may assure certain functional properties such as proper termination, thus giving starting points for narrowing down the possible interactions. For approximation of a service's costs we propose a static analysis approach on a constrained set of interaction partners.

For two services A and B , question is: How much does it cost the provider of A to interact with B ? As a prerequisite, we assume that each action a of A has a distinct cost. However prior to execution, it is impossible to give one discrete overall cost: Both A and B may contain non-determinism resulting in different runs of their interaction. So we propose an *interval* for the overall cost instead. Naively, we can explore the whole state space to determine the minimal and maximal overall costs, taking only into account the maximal runs, that is those that are either infinite or end in a deadlock state. This however can be very complex and may be impossible, if the state space is infinite or too big to fit into memory.

We thus propose a simplification of the previous question: If both A and B terminate, how much did the interaction cost the provider of A ? If we further assume that termination always means reaching a final state ω of A , the question boils down to: How much does it cost the provider of A to reach ω from its initial state α when A and B are coupled? We identify three core challenges:

(i) As for the overall cost, the cost of reaching ω is not one distinct number but a cost interval $C = (c_{\min}, c_{\max})$, because there typically exists not only one, but a set R of runs from α to ω , which can even be infinite. (ii) We find the nature of R not only being dependent on the behavior of A , but also on the behavior of B . (iii) ω is not necessarily reachable in the interaction of A and B , resulting in an undefined cost interval.

In this paper, we will tackle those three challenges with classical Petri net analysis. We translate the model of the composition $A \oplus B$ into a system of linear equations – the *state equation* [2] – and use *linear optimization* to minimize and maximize costs. The resulting bounds $X = (x_{\min}, x_{\max})$ will over-approximate C . Although we lose some precision with this technique, we expect the proposed approach to make cost estimation before execution feasible in practice.

If B is not given as a model, but only as a set Ψ of *constraints*, we build the state equation of A alone and augment it by the constraints in Ψ . Again, we use linear optimization which yields a valid over-approximation $X' = (x'_{\min}, x'_{\max})$ of C . We observe that X' is not as tight as X : X' is valid for a whole set of partner services B_Ψ (induced by Ψ) in contrast to only one explicitly given partner service B . However, computation of X' can be done in a less time critical phase, whereas X is computed after B has been selected and also requires a complete behavioral model of B . Furthermore, we attend the special case that $\Psi = \emptyset$, resulting in the complete set of partners of A , similarly to the work in [3].

The third challenge is harder to come by. In this paper we aim at ensuring, that *if* A reaches ω , *then* the found cost interval applies. The found cost intervals for each final marking can be *composed* for a global overview over costs. To avoid complexity from this source, we draft a method to directly find minimal and maximal costs for given sets of final markings.

The rest of the paper is structured as follows: In Sect. 2 we introduce the necessary formal notions. Section 3 describes the approach itself together with the above announced specializations and generalizations. Finally we will conclude and will give an outlook for application.

2 Notation

Let $f : A \rightarrow B$ be a function, then for $A' \subseteq A$, $f(A')$ is defined as $\{f(a) \mid a \in A'\}$. For the rest of this paper, let us assume a set \mathcal{C} of *message channels* and message exchange to be *asynchronous*, messages may even overtake each other. Sending a message over channel a is denoted by $!a$ and receiving a message over channel a is denoted by $?a$. Then $\mathcal{E} = \{!c \mid c \in \mathcal{C}\} \cup \{?c \mid c \in \mathcal{C}\}$ denotes the set of all *sending* and *receiving events*, respectively. Furthermore, each service uses a channel in at most one direction: sending or receiving (viz. a service cannot unsend a message).

Now an *open net* $N = (P, T, F, \alpha, \Omega, ev)$ is a classical P/T-net [4] (P, T, F, α) . $ev : T \rightarrow (\mathcal{E} \cup \{\tau\})$ is a *labeling function* indicating for each transition t if it is either communicating ($ev(t) \in \mathcal{E}$ with $ev(t) = ?c \Rightarrow !c \notin ev(T)$) or internal to the net ($ev(t) = \tau$). Additionally we define a set of final markings Ω , which indicate proper termination. The *interface* $\mathcal{E}(N) = ev(T) \setminus \{\tau\}$ of an open net N comprises all labels used by the transitions of N . Examples for open nets are depicted in Figs. 1 and 2. As

usual, places are depicted by circles, transitions by rectangles and the flow relation by arcs. Tokens are black dots, labels are written inside the transitions, omitting τ .

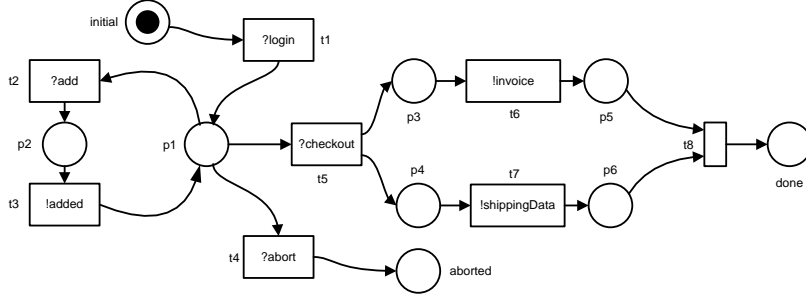


Fig. 1. Open net shop with $\Omega = \{[aborted], [done]\}$

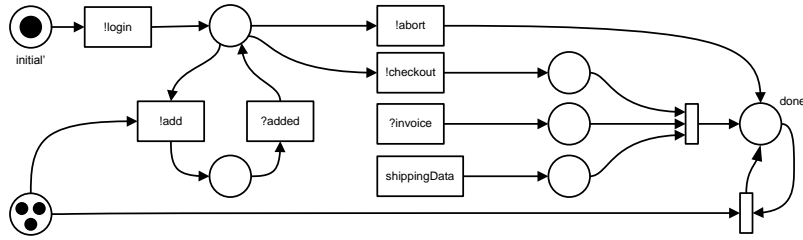


Fig. 2. Open net customer with $\Omega = \{[done']\}$

The open net shop models an online shop. After logging in, a customer can decide to add products by sending messages via channel add which are confirmed via added. The customer can either decide to send an abort message or to checkout. After receiving a message via checkout, the shop sends information about shipping and an invoice. The two final markings $[done]$ and $[aborted]$ specify the two expected results of the interaction. The open net customer is a *partner* of shop: Their interfaces are compatible.

We assume the standard firing semantics for transitions, thus $\mu \xrightarrow{t} \mu'$ means a *step* from μ to μ' by *firing* t . A transition sequence $t_0 t_1 t_2 \dots$ is called *firing sequence* of N if there exists a sequence of markings $\mu_0 \mu_1 \mu_2 \dots$, such that $\mu_0 = \alpha$ and for each $i = 0, 1, 2, \dots$, $\mu_i \xrightarrow{t_i} \mu_{i+1}$ is a step of N . Note that for each firing sequence, there exists exactly one corresponding sequence of markings. We thus say that a finite firing sequence $t_0 t_1 t_2 \dots t_n$ *ends* in μ_{n+1} . We call a finite firing sequence *terminating*, if it ends in a final marking. The Parikh vector [5] of a transition sequence r is denoted as $occ(r)$. $r|_{T'}$ denotes the restriction of r to elements of T' , which we canonically extend

to Parikh vectors. The *behavior* of an open net is the set of all its firing sequences, denoted as $\text{Beh}(N)$. The set of all firing sequences ending in a marking μ is denoted as $\text{Beh}(N, \mu)$.

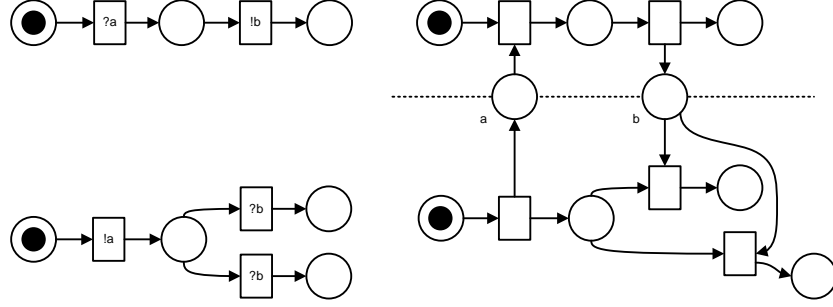


Fig. 3. Two open nets and their composition

The *composition* of two partners N and N' to the open net $N \oplus N'$ is realized by introducing buffer places and corresponding arcs as depicted in Fig. 3. Inspecting the structure of $N \oplus N'$, we find that a firing sequence of $N \oplus N'$ corresponds to one firing sequence of N as well as one of N' , allowing us to analyze N in isolation and draw conclusions for the behavior of N in $N \oplus N'$.

Thus, let N, N' be partners and μ, μ' be markings of N, N' respectively. Then, $r \in \text{Beh}(N \oplus N', \mu + \mu')$ implies $r|_T \in \text{Beh}(N, \mu)$ and $r|_{T'} \in \text{Beh}(N', \mu')$.

We now introduce *costs* for actions of services into our formal model. Each transition t has a *globally fixed integer cost*, denoted as $\text{cost}(t)$. The cost of a transition sequence only depends on its Parikh vector: $\text{cost}(r) = \text{cost}(\text{occ}(r)) = \sum_{t \in T} \text{occ}(r)(t) \cdot \text{cost}(t)$. Let μ be a marking of an open net N . If $\text{Beh}(N, \mu) \neq \emptyset$, then the minimal and maximal cost of μ in N are defined as $\min(\{\text{cost}(r) \mid r \in \text{Beh}(N, \mu)\})$ and $\max(\{\text{cost}(r) \mid r \in \text{Beh}(N, \mu)\})$, denoted as $\perp(N, \mu)$ and $\top(N, \mu)$ respectively. Otherwise, the minimal and maximal cost are undefined. Let $T' \subseteq T$, then $\perp(N, \mu)|_{T'}$ and $\top(N, \mu)|_{T'}$ denote the minimal and maximal cost for μ by only taking into account transitions in T' .

As an example, consider that the transitions t_1, \dots, t_8 of the open net shop in Fig. 1 have costs of 10, 5, 1, 10, 5, 20, 20, 1 respectively. Then, the firing sequence $t_1 t_2 t_3 t_2 t_3 t_4$ has a cost of 32. For marking $[p_3]$ the cost interval is $(15, \infty)$.

3 Cost estimation

The *state equation* [2] is a proven tool in Petri net analysis: Let as usual $\bullet x (x \bullet)$ denote the *preset (postset)* of a node $x \in P \cup T$. The $P \times T$ -matrix I_N , the *incidence matrix* of N , is defined as follows: $I_n(p, t) = -1$ if $p \in \bullet t \setminus t \bullet$, $I_n(p, t) = 1$ if $p \in t \bullet \setminus \bullet t$ and $I_n(p, t) = 0$, otherwise. Let y be a vector with $y(p) = \mu(p) - \alpha(p)$, $p \in P$. Then, $X(N, \mu)$ denotes *all non-negative integer solutions* of the state equation $I_N \cdot x = y$ of N

with respect to marking μ . The state equation for the open net shop in Fig. 1 is displayed in Table 1.

Table 1. State equation of the open net shop (Fig 1) w.r.t. marking [done].

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	y
initial	-1								-1
p_1	+1			-1	-1				
p_2		+1	-1						
p_3					+1	-1			
p_4					+1		-1		
p_5						+1		-1	
p_6							+1	-1	
done								+1	+1
abort				+1					

The Parikh vector of any firing sequence ending in μ is a solution of the state equation. As an example, consider open net shop in Fig. 1: $t_1 t_2 t_3 t_2 t_3 t_5 t_6 t_7 t_8$ is a firing sequence ending in the marking [done]. Its Parikh vector is $(1\ 2\ 2\ 0\ 1\ 1\ 1\ 1)$, which is a solution of the state equation. The reverse does not hold for the general case: By solving the state equation tokens can be "borrowed" from and "given back" to places, leading to an effect of 0, although no firing sequence with this solution as a Parikh vector exists. Obviously, $X(N, \mu)$ is a valid over-approximation of $\text{occ}(\text{Beh}(N, \mu))$. Because the cost of a firing sequence is only dependent on its Parikh vector, the state equation is a valid mechanism to estimate costs for a given marking.

In this section, we present two approaches: First, we show how for *two given services* the costs for one provider can be estimated by solving the state equation of the composition of their models. Then, we explain how this approach can be generalized to the setting where the second partner is only represented by a set of *constraints*. For the second approach, we can shift the analysis effort to a time-uncritical phase. Ensuing, we have a critical look at both approaches, collecting results for the open net shop in Fig. 1. Finally, we draft how our general approach can be extended from estimating costs for a given state to a given (even infinite) set of states.

3.1 Estimating costs for one provider in a specific composite

In the following let N (with its transition set T) and N' be partners and ω, ω' be final markings of N and N' , respectively. From the property of the state equation follows:

Lemma 1. $\text{occ}(\text{Beh}(N \oplus N', \omega + \omega'))|_T \subseteq X(N \oplus N', \mu)|_T$.

Directly from set algebra follows that the state equation of $N \oplus N'$ provides a *valid over-approximation* for the costs of the provider of N in $N \oplus N'$:

Theorem 1. $\text{Beh}(N \oplus N', \omega + \omega') \neq \emptyset$ implies $\min(\text{cost}(X(N \oplus N', \omega + \omega')|_T)) \leq \perp(N \oplus N', \omega + \omega')|_T \leq \top(N \oplus N', \omega + \omega')|_T \leq \max(\text{cost}(X(N \oplus N', \omega + \omega')|_T))$.

Thus, the costs for the provider of N can be estimated by solving the linear problem for each final marking of $\Omega \times \Omega'$. From our experience, the number of final markings is very small, if not 1. As an example, evaluating the state equation for the composite $\text{shop} \oplus \text{customer}$, we find two cost intervals: $(56, 74)$ for the final marking $[\text{done} + \text{done}']$ and $(20, 38)$ for $[\text{aborted} + \text{done}']$, which we can compose to an *overall cost estimation* of $(20, 74)$. We observe, that this is the actual cost interval. Obviously, the approach is symmetrical: The costs for the provider of N' can be estimated analogously. We will now generalize our approach from a completely known open net N' to a set of constraints, describing a set of services.

3.2 Pre-estimating costs for one provider

Assume now that not an open net N' is given but a *set of constraints* of the following syntax and semantics: A constraint is an *inequality* or *equation* $l * r$ where $*$ $\in \{\leq, \geq, =\}$, consisting of an *integer linear combination* l of *event labels* and an *integer value* r . An open net N' *solves* $l * r$ if for any terminating firing sequence in N' , its event occurrence vector solves $l * r$. N' *solves* the set Ψ of constraints, if N' solves each $\psi \in \Psi$. We denote the set of open nets solving Ψ with \mathcal{B}_Ψ . As an example, the open net shop in Fig. 1 solves the constraints $?\text{login} = 1$ and $?\text{add} - !\text{added} = 0$. [3] shows how such a set of constraints can be obtained from an open net by static analysis.

Intuitively, we take the state equation of N and *augment* it correspondingly with the given set of constraints Ψ : Because every message has been consumed in a final marking of the composite, we find that for each sent message of a partner of N , N has received the message and vice versa. Thus, for each message x sent (received) by N' , a transition receiving (sending) x of N has fired once. This induces a set of constraints $\widehat{\Psi}(N) = \{\sum_{t \in T} l(\text{ev}(t)) \cdot t * r \mid l * r \in \Psi\}$ on the transition occurrence of N . As an example, consider the open net shop in Fig. 1 and assume $\Psi = \{!\text{add} \leq 10\}$ as well as $N' \in \mathcal{B}_\Psi$. Then, in any terminating firing sequence of $N \oplus N'$ the only transition with label $?\text{add}$, namely t_2 fires only up to 10 times, thus $\widehat{\Psi}(N) = \{t_2 \leq 10\}$.

Therefore, the state equation of N with respect to ω together with $\widehat{\Psi}(N)$, its solutions denoted as $X_\Psi(N, \omega)$, over-approximates all firing sequences ending in $\omega + \omega$ of $N \oplus N'$ if $N' \in \mathcal{B}_\Psi$.

Lemma 2. $N' \in \mathcal{B}_\Psi$ implies $\text{Beh}(N \oplus N', \omega + \omega')|_T \subseteq X_\Psi(N, \omega)$.

Analogously to our approach with full knowledge of N' , we can conclude that for a given Ψ , the costs for the provider of N for N interacting with an $N' \in \mathcal{B}_\Psi$ can be *over-approximated* by using the state equation:

Theorem 2. $N' \in \mathcal{B}_\Psi$ and $\text{Beh}(N \oplus N', \omega + \omega') \neq \emptyset$ implies $\min(\text{cost}(X_\Psi(N, \omega))) \leq \perp(N \oplus N', \omega)|_T \leq \top(N \oplus N', \omega)|_T \leq \max(\text{cost}(X_\Psi(N, \omega)))$.

Thus, solving the linear program for each $\omega \in \Omega$ yields a *valid cost estimation*. Expecting $|\Omega|$ to be small, this approach seems feasible for use in practice. Furthermore, we draft in Sect. 3.4 how a valid cost estimation can directly be computed for a set of final markings.

3.3 A critical look

We have collected a small number of results in Table 2. The table shows for the open net shop in Fig. 1 how the *actual* cost intervals (c_{\min}, c_{\max}) and the *estimated* cost intervals (x_{\min}, x_{\max}) for a selected final marking ω correspond for partners specified by Ψ . The results were gained manually; there does not exist an implementation yet. In the example, the estimated cost interval is identical with the actual cost interval for any Ψ . We also see that open net customer in Fig. 2 solves $\{1 \leq !\text{add} \leq 10\}$. Composing the second and fifth row in the table, we find an estimated cost interval of (20, 116), which is not as tight as the former result (20, 74), but still valid.

In the general case however, results are not necessarily this precise. A factor are *t-invariants*, transition vectors x , such that $I_N \cdot x = 0$. For the open net shop in Fig. 1, $(0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0)$ is a t-invariant. Obviously, if y is a solution of state equation S and x is a t-invariant, then for any $n \in \mathbb{N}$ holds that $y + n \cdot x$ is also a solution of S . One can create an example with a t-invariant that never fires, intuitively an unmarked loop structure. Thus, if a t-invariant x exists and $\text{cost}(x) > 0$ ($\text{cost}(x) < 0$), cost estimation yields unbounded for the upper (lower) bound, unless there is an additional constraint bounding it. For the t-invariant $x = (0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0)$, $\text{cost}(x)$ is greater than zero. *Constraining* it through the constraint set $\{1 \leq !\text{add} \leq 10\}$ bounds the cost interval because x can not be added arbitrarily often due to bounding the occurrence t_2 . It is up to future work to find a way to handle t-invariants and to use them as a starting point for narrowing down interesting Ψ .

Table 2. Minimal/maximal costs for the provider of open net shop (Fig. 1)

Ψ	ω	c_{\min}	c_{\max}	x_{\min}	x_{\max}
{}	[done]	56	∞	56	unbounded
$\{0 \leq !\text{add} \leq 10\}$	[done]	56	116	56	116
$\{!\text{abort} = 1\}$	[done]	undefined	undefined	infeasible	infeasible
{}	[aborted]	20	∞	20	unbounded
$\{0 \leq !\text{add} \leq 10\}$	[aborted]	20	80	20	80
$\{!\text{abort} = 1\}$	[aborted]	20	∞	20	unbounded

3.4 Costs for sets of markings

Solving one linear program for *each* $\omega \in \Omega$ is not feasible for big or even infinite Ω . For the case that we can express or approximate the given set of markings as a *set of linear constraints*, we build a system of linear inequalities similar to the state equation and use the same techniques as described above.

4 Conclusion

In this paper, we presented an approach to *estimate* the costs for the provider of a service A for interacting with a service B . Thereby, we concentrated on two scenarios: Either

the partner B is known in detail, then the estimation process boils down to static analysis of the *composed* system $A \oplus B$. Or, B is not known and only narrowed down by a set of *constraints* Ψ . In this case, we analyze the behavior of A in *isolation*, constraining it according to Ψ . The downside of static analysis is a loss of precision, for which we identify t-invariants as an important factor. Still, we find that our approach tackles the first and the second challenge introduced in Sect. 1. For the case of a not reachable final marking, our approach might yield a cost interval although it is not defined. We consider this a low price to pay in contrast to the state space explosion problem.

5 Related and future work

In [6, 7] the authors analyze BPMN [8] models. For minimal and maximal costs [6] uses the *Dijkstra* algorithm with complexity $O(n \log n)$. Our approach is based on the *simplex* algorithm which is known to normally have linear runtime, although worst-case complexity is exponential. In the case of acyclic services, the state equation approach is even sufficient, so we also find strict bounds. The approach in [7] is pattern-based, an approach not applicable to arbitrary graph-based formalisms such as BPMN or open nets.

In this paper, we only considered a single fixed cost for each atomic action of a service. As a start it is intuitive that a certain step has always the same costs. A natural extension would be allow intervals [6] or even stochastic values [9]. However, as long as the cost functions are still linear, the extension is canonical. Additionally, costs might be history-dependent. An action might have some fixed and some execution costs, such that a consecutive execution of this action might be less expensive. Furthermore it would be interesting to use the approach as a decision help for service discovery. As a base we can imagine a cost profile stored in a service repository, including cost estimations and acceptable cost intervals, inducing a concept of compatibility under costs.

References

1. Papazoglou, M.P.: Web Services: Principles and Technology. Pearson - Prentice Hall, Essex (July 2007)
2. Lautenbach, K.: Liveness in Petri Nets. St. Augustin: Gesellschaft für Mathematik und Datenverarbeitung Bonn, Interner Bericht ISF-75-02.1 (1975)
3. Sürmeli, J.: Profiling services with static analysis. In Freytag, T., Eckleder, A., eds.: AWPN. (2009)
4. Reisig, W.: Petri Nets: An Introduction. Volume 4 of Monographs in Theoretical Computer Science. An EATCS Series. Springer (1985)
5. Parikh, R.: On context-free languages. J. ACM **13**(4) (1966) 570–581
6. Magnani, M., Montesi, D.: BPMN: How much does it cost? An incremental approach. In: BPM. (2007) 80–87
7. Sampath, P., Wirsing, M.: Computing the cost of business processes. In: UNISCON. (2009) 178–183
8. OMG: Business Process Model and Notation 2.0. (August 2009)
9. van Hee, K.M., Verbeek, H.M.W., Stahl, C., Sidorova, N.: A framework for linking and pricing no-cure-no-pay services. T. Petri Nets and Other Models of Concurrency **2** (2009) 192–207