An efficient necessary condition for compatibility

Olivia Oanea* and Karsten Wolf

Universität Rostock, Institut für Informatik
18051 Rostock Germany
{olivia.oanea,karsten.wolf}@uni-rostock.de

Abstract. Composing services makes sense only if they are compatible, i.e. composition does not lead to problems such as livelocks or deadlocks. In general, compatibility can be checked using state space explorations on any kind of formal models of services.
Petri nets, one of the formal models in use, offer a rich theory for reasoning without exploring a state space. Among the techniques is the so-called *state equation* which forms a linear algebraic necessary condition for reachability of states.
In this article, we show how the state equation can be applied for a necessary condition for compatibility. This way, the number of expensive state space based compatibility checks can be drastically reduced. The condition can be applied even if compatibility is achieved through the construction of a behavioral adapter (mediator).

1 Introduction

Service behaviors are compatible if their composition forms a closed system (every outbound channel of a service is merged to an inbound channel of some other service) and all involved services can execute their control flow completely. Compatibility can be augmented with the requirement that all or certain activities in the participating services can occur or other semantical constraints.

In this paper we show an approach for alleviating the costs of the compatibility check for services modeled with Petri nets using their state equation. The state equation provides a necessary condition for reachability of the final states of the services in the composition under several constraints such as the enabling of some events or choice covering. This result can be applied directly to adapter synthesis [1]. Service adaptation (mediation) is a semi-automatic approach of correcting incompatibilities between services in which transformation rules are provided normally by hand to correct the message flow. The state equation provides a necessary condition for the existence of such an adapter that uses the specified rules.

In the remainder of this article, we first introduce notations for Petri net models for services and the state equation. Section 3 gives the necessary conditions for compatibility and derives other necessary conditions for compatibility under some additional constraints. Section 4 presents a necessary condition to adaptability. Section 5 concludes the paper.

* Supported by German Research Foundation (DFG) under grant WO 1466/11-1

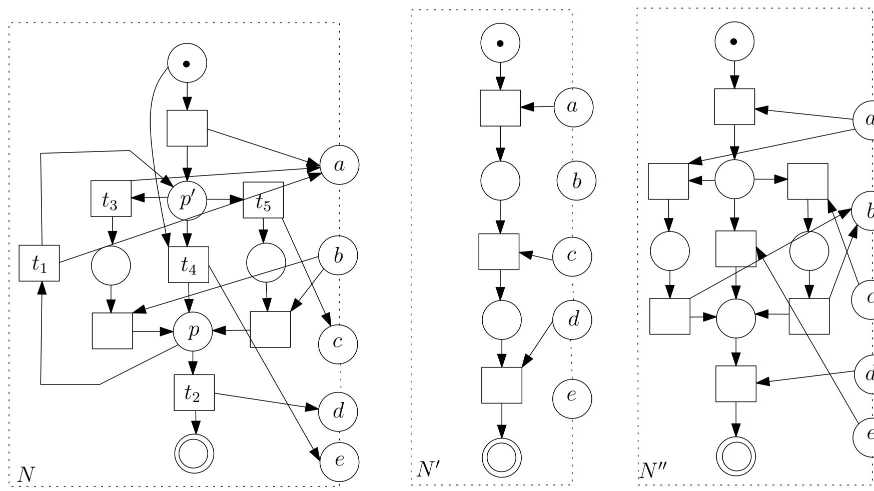

Fig. 1: An open net N and its partner open nets N' and N''

2 Petri nets as models of services and the state equation

Let $\Sigma = \{a, b, c, \dots\}$ be a finite message type set, $?\Sigma = \{?a, ?b, ?c \dots\}$ a finite set of receive events, and $!\Sigma = \{!a, !b, !c \dots\}$ a finite set of send events. We also write $\overline{?\Sigma} = !\Sigma$ and $\overline{\overline{?\Sigma}} = ?\Sigma$.

We consider services modeled as *open nets*. An open net [2] is a Petri net [3] with a special set of interface places which represent the communication channels with other nets.

Definition 1. *An* open net *is a tuple* $N = (P \cup P_i \cup P_o, T, F, m_0, M_f, l)$, *where*

- P, P_i, P_o *are the pairwise disjunct finite sets of internal/input/output places;*
- T *is the finite set of transitions so that* $(P \cup P_i \cup P_o) \cap T = \emptyset$ *which are labeled by the partial function* $l \colon T \to ?\Sigma \cup !\Sigma$*;*
- $F \colon ((P \cup P_i \cup P_o) \times T) \cup (T \times (P \cup P_i \cup P_o)) \to \mathbb{N}$ *represents the flow function so that* $F(p, t) = F(t', p') = 0$, *for all* $(p, t) \in P_o \times T$ *and* $(t', p') \in T \times P_i$*;*
- m_0, M_f *represent the initial state (marking) and the finite set of final states, respectively. We consider states as vectors over the set of places.*

An open net is called closed when its interface is empty, i.e. $P_i \cup P_o = \emptyset$. The projection of an open net on its transitions and internal places is a closed net denoted by $\widehat{N}$. Open nets over $?\Sigma \cup !\Sigma$ are composed [2] by merging their interface places (i.e. an input and with an output place denoting the same message channel) and is denoted by $\oplus$, with the corresponding initial and final markings. Figure 1 shows three open nets N, N', N'', each with the final marking with a token on its double circled place.

A transition $t \in T$ is *enabled* in a marking m if $F(p, t) \le m(p)$ for all places p. An enabled transition may fire yielding a (reachable) marking m' so that

$m'(p) = m(p) - F(p,t) + F(t,p)$ for all places p, which is denoted by $m \xrightarrow{t} m'$. The reachability relation can be extended to sequences of transitions $\sigma \in T^*$, which is denoted by $\xrightarrow{\sigma}$. Two open nets are called *compatible* if their composition weakly terminates, i.e. from each state reachable from the initial state of the composition, it is possible to reach a final state of the composition. A weaker notion of compatibility is deadlock-freedom, i.e. at each non-final reachable state (in the composition) it is possible to fire a transition.

Reachability analysis for Petri nets can be achieved by using typical structural methods, e.g. methods which find algebraic approximations of the state space with finite representation. The *state equation* [4] relates the behavior of a net (given by states and firing sequences) and its structure (incidence matrix) and can be solved using standard linear programming [5].

The *incidence matrix* $C_N \in \mathbb{N}^{(P \cup P_i \cup P_o) \times T}$ is defined by $C_N(p,t) = F(t,p) - F(p,t)$ for all $(p,t) \in (P \cup P_i \cup P_o) \times T$. Let $\sigma \in T^*$ be transition sequence. The Parikh vector of σ is a vector $\bar{\sigma} \in \mathbb{N}^T$ which assigns to each transition $t \in T$ its number of occurrences in σ. Let $\bar{\sigma}(a) = \sum_{t \in T : l(t) = a} \bar{\sigma}(a)$ denote the number of occurrences of all transitions labeled by $a \in !\Sigma \cup ?\Sigma$. Given a firing sequence $m \xrightarrow{\sigma} m'$ of N, the firing equations for all places of N and all transitions in σ can be written in matrix form $m' = m + C \cdot \bar{\sigma}$, which is called the *state equation*.

Proposition 1 (Necessary condition for reachability). *For every finite firing sequence $m \xrightarrow{\sigma} m'$ of N, the state equation $m' = m + C_N \cdot \bar{\sigma}$ holds.*

3 Necessary condition for compatibility

We state now a necessary condition for compatibility as weak termination of two composed open nets. The first conditions represent the state equations of the open nets without their interface places. The last condition means that in all solutions to the equation the number of occurrences of receiving events should be equal to the number of occurrences for sending events for each such event.

Corollary 1. *If N and N' are compatible (w.r.t. weak termination), then the system $LP(C_{\widehat{N}}, C_{\widehat{N}'}, m_0, m_0', m_f, m_f', x, x')$ is feasible.*

$$LP(C_{\widehat{N}}, C_{\widehat{N}'}, m_0, m_0', m_f, m_f', x, x') : \quad \begin{array}{rl} m_f = m_0 + C_{\widehat{N}} \cdot x & x \in \mathbb{N}^T \\ m_f' = m_0' + C_{\widehat{N}'} \cdot x' & x' \in \mathbb{N}^{T'} \\ x(a) = x'(\bar{a}) & \forall a \in ?\Sigma \cup !\Sigma \end{array}$$

If the equation does not have any solution then the final marking will not be reachable in the composition from the initial marking.

Remark 1. In case services have more final states, separate systems of equations are solved for each possible combination. For the nets N and N' in Figure 1 $LP(C_{\widehat{N}}, C_{\widehat{N}'}, m_0, m_0', m_f, m_f', x, x')$ does not have any solution. Therefore, N and N' are incompatible. Note that the converse does not hold, e.g. the nets N and N'' in Figure 1, $x''(?a) = x(!a) = 2$, $x(?b) = x''(!b) = 1$, $x(!d) = x''(?d) = 1$, $x(!c) = x''(?c) = 0$ and $x(!e) = x''(?e) = 0$ is a solution for $LP(C_{\widehat{N}}, C_{\widehat{N}''}, m_0, m_0'', m_f, m_f'', x, x'')$, however N and N'' are incompatible as we shall see in the remainder.

If $N \oplus N'$ is deadlock-free then at each non-final reachable marking in the composition there is an enabled transition, i.e. adding the disabling condition for each transition leads to an infeasible system.

Corollary 2 (deadlock-freedom). *If $N \oplus N'$ is deadlock-free then the following system of inequations has no solution:*

$$\begin{array}{rl} m = m_0 + C_{\widehat{N}} \cdot x & x \in \mathbb{N}^T, m \in \mathbb{N}^P \\ m' = m'_0 + C_{\widehat{N}'} \cdot x' & x' \in \mathbb{N}^{T'} m' \in \mathbb{N}^{P'} \\ x(a) = x'(\bar{a}) + m''(p_a) & \forall a \in ?\Sigma \cup !\Sigma \\ m <> m_f \wedge m' <> m'_f \wedge m'' <> 0^{P_i \cup P_o} & m'' \in \mathbb{N}^{P_i \cup P_o} \\ \bigvee_{p: F_{N \oplus N'}(p,t) > 0} ((m + m' + m'')(p) < F_{N \oplus N'}(p,t)) & \forall t \in T \cup T' \end{array}$$

3.1 Necessary conditions for compatibility under constraints

Several variations for compatibility notions have been introduced [6–8] which define behavioral constraints which can imposed on interacting services. Among these settings we mention transition cover and place cover.

Message and event cover

Definition 2. *We call an action a in $?\Sigma \cup !\Sigma$* covered locally/globally *iff a transition/all transitions labeled by a in the composition eventually becomes enabled in the composition. A message place (channel) $p \in P_i \cup P_o$ is called covered if $m(p) > 0$, for some reachable marking m in the composition.*

Let N and N' be two open nets and $a \in ?\Sigma \cup !\Sigma$. We state now conditions which should be added to $LP(C_{\widehat{N}}, C_{\widehat{N}'}, m_0, m'_0, m_f, m'_f, x, x')$ to enforce local, global event cover, place and message cover.

local event cover $x(t) > 0$ ($t \in T : l(t) = a$) or $x'(t') > 0$ ($t' \cup T' : l(t) = a$);
place cover for $p \in P$ there exists a $t \in T$ so that $F(p,t) > 0$ and $x(t) > 0$ (similarly if $p \in P'$);
global event cover $x(t) > 0$, for all $t \in T : l(t) = a$ or $x'(a) > 0$;
message channel cover $x(a) > 0$ and $x'(\bar{a}) > 0$.

In $N \oplus N''$ in Figure 1, a is locally covered but not globally covered (transition t_1). The message channel e is covered neither in $N \oplus N'$ nor in $N \oplus N''$.

Free-choice sending cover Here, we want to strengthen the previously stated condition by taking into account that compatibility does not refer to a single execution (as the state equation would suggest). If an execution passes an internal decision of one service then its partner needs to be able to react to all possible outcomes for this decision. With the following consideration, we want to incorporate this observation into our condition at least for so-called free-choice decisions [3].

Let $x \in P \cup T$. The conflict cluster $\nu(x)$ of x is the smallest set satisfying (1): $x \in \nu(x)$, (2): $\forall q \in T\colon {}^\bullet q \cap \nu(x) \neq \emptyset \implies q \in \nu(x)$ and (3): $\forall q \in P\colon q^\bullet \cap \nu(x) \neq \emptyset \implies q \in \nu(x)$. We write ν when x is clear from the context. A conflict cluster $\nu(x)$ so that $|\nu(x)| > 2$ is called a sending free-choice conflict cluster (SC) iff for all $t_1, t_2 \in \nu \cap T$, ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$ implies ${}^\bullet t_1 = {}^\bullet t_2$ and $l(t) \in !\Sigma$ for all $t \in T \cap \nu$. In Figure 1 $\{p, t_1, t_2\}$ represents such a SC in N. Note that a SC in $\widehat{N}$ is also a SC in N.

A SC in the composition of two nets N and N' is called covered if each transition of the SC is in some firing sequence from the initial marking to the final marking of the composition. For compatible partners, every reachable SC in a service should be resolved by the partner.

Corollary 3. *Let ν be a SC with $\nu \cap T = \{t_1, t_2\}$ in N. If N and N' are compatible and ν is covered in $N \oplus N'$, then $CLP(C_{\widehat{N}}, C_{\widehat{N}'}, \nu)$ is feasible.*

$$CLP(C_{\widehat{N}}, C_{\widehat{N}'}, \nu)\colon \quad \begin{array}{l} LP(C_{\widehat{N}}, C_{\widehat{N}'}, m_0, m'_0, m_f, m'_f, x, x') \\ LP(C_{\widehat{N}}, C_{\widehat{N}'}, m_0, m'_0, m_f, m'_f, \bar{x}, \bar{x}') \\ x(t_1) > 0 \wedge x'(t_2) > 0 \\ \{\nu' \ SC \text{ in } \widehat{N}' | \nu' \cap T' = \{t'_1, t'_2\} \wedge \bar{x}(t'_1) > 0 \wedge \bar{x}'(t'_2) > 0 \wedge \\ \quad \wedge l'(t'_1) = \overline{l(t_1)} \wedge l'(t'_2) = \overline{l(t_2)}\} \end{array}$$

The last condition checks for the existence of a conflict cluster ν' receiving the messages sent by ν. The open nets N and N'' in Figure 1 are incompatible as $CLP(C_{\widehat{N}}, C_{\widehat{N}'}, \nu)$ has no solutions (the choice between the transitions labeled by $!a$ and $!d$ in N'' is not covered) even if $LP(C_{\widehat{N}}, C_{\widehat{N}''}, m_0, m''_0, m_f, m''_f, x, x'')$ has solutions.

Remark 2 (deadlock-freedom under constraints cover). We can relax the deadlock-freedom condition in Corollary 2 to express a necessary condition for local event (transition) cover and SC cover:

t **cover** $\bigvee_{p:F_{N\oplus N}(p,t)>0} (m + m' + m'')(p) \geq F_{N\oplus N'}(p, t)$, where $t \in T \cup T'$;
SC **cover** $\bigvee_{p:F_{N\oplus N}(p,t)>0} (m + m' + m'')(p) \geq F_{N\oplus N'}(p, t)$ for all $t \in \nu$.

Remark 3 (behavioral SC). The transition t_4 of N in Figure 1 is dead and removing it from N does not influence compatibility of N with any other partner. Hence we can consider "behavioral" SC's (e.g. $\{p', t_3, t_5\}$) to be checked for cover.

4 Necessary condition for adapter synthesis

The open nets N_1 and N_2 in Figure 2 do not satisfy the necessary condition in Corollary 1, hence they are incompatible. Adapters are used to solve incompatibilities between interacting services. We consider here the approach in [1] with weak termination as compatibility notion, where adapters are partially specified by transformation rules on messages called SEA (Specification of Elementary Actions). A general rule is described by $r\colon x \mapsto x'$, where $x \in \mathbb{N}^{!\Sigma}$ and $x' \in \mathbb{N}^{?\Sigma}$.

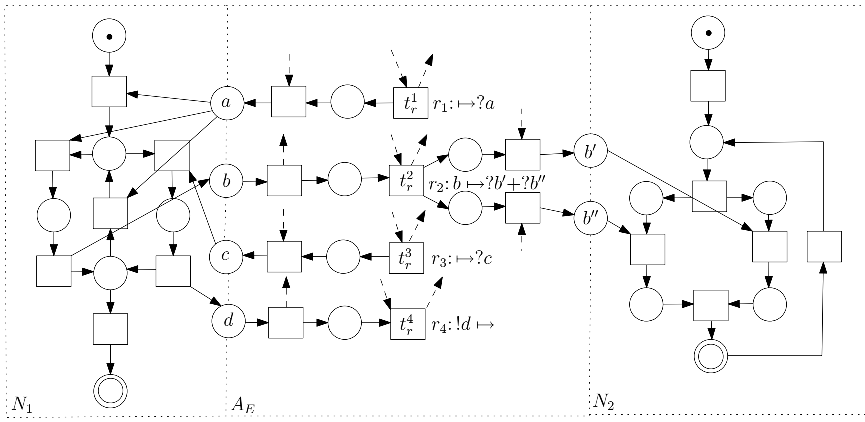

Fig. 2: Two open nets N_1 and N_2 and their partial adapter A_E

The example in Figure 2 shows typical transformation rules: creation of a message (e.g. $!d \mapsto$), deletion of a message ($\mapsto ?c$), splitting a message into two messages ($!b \mapsto ?b' + ?b''$). Each transformation rule is transformed into an open net which communicates with the initial services and with an entity which controls the application of these rules (e.g. the transition t_r^1) and the sending/receiving of messages (denoted by dashed arrows). The open net obtained from the transformation rules is called partial adapter A_E. The adapter synthesis procedure computes a partner C which controls $N_1 \oplus A_E \oplus N_2$ and the final adapter is $C \oplus A_E$.

A direct consequence of Corollary 1 is that compatible partners have a solution to their own state equation. We state this condition for the adapter setting.

Corollary 4. *If N_1 and N_2 are adaptable by the set of transformation rules R, then the state equation for $N_1 \widehat{\oplus A_E \oplus} N_2$ with initial marking $m_0^1 + m_0^2$ and final marking $m_f^1 + m_f^2$ holds.*

The state equation for $N_1 \widehat{\oplus A_E \oplus} N_2$, where A_E is the partial adapter for the rules $\{r_1, r_3, r_4\}$, does not yield any solution, thus N_1 and N_2 are not adaptable by $\{r_1, r_3, r_4\}$.

In addition, we can formulate a necessary condition for transformation rule cover. Let $r\colon \sigma \longrightarrow \sigma'$. We add to the state equation of $N_1 \widehat{\oplus A \oplus} N_2$ the constraint $x(t_r) > 0$, where t_r is the transition corresponding to the application of the rule. Thus, we can eliminate rules which will never be fired in conjunction with a proper terminating execution. In Figure 2, r_3 and r_4 are redundant rules.

5 Conclusion

In this paper we stated some necessary conditions for service compatibility using the state equation for Petri nets. The advantage of using this approach to state space methods (e.g. [9–11]) is its lower computational complexity [5] (polynomial for real solutions/exponential in the worst case for integer solutions). An area of application for this approach is service discovery and service composition [8, 2], i.e. finding well-behaved partners for a particular service in a repository of services. Service discovery and composition are inherently costly job (both from time and space) w.r.t. the size of the repository and of the services themselves. Using such a quick check can ease the task of a broker for discovering/adapting potentially compatible partners for a service by disposing of those services which do not satisfy the necessary criterion.

The approach presented in this paper allows for (in)compatibility to be analyzed in a compositional way (incorrectness of a component can be used to derive the incorrectness of the composition). This is complementary to structural methods used in soundness analysis [12, 13] of monolithic workflow. As future work, we plan to implement the state equation approach as a preliminary check for service composition and adaptability and evaluate the efficiency of this approach in the large on a set of case studies provided by industry.

References

1. Gierds, C., Mooij, A.J., Wolf, K.: Specifying and generating behavioral service adapter based on transformation rules. Technical Report CS-02-08, Universität Rostock, Rostock, Germany (2008)
2. Wolf, K.: Does my service have partners? LNCS ToPNoC **5460**(II) (2009) 152–171 Special Issue on Concurrency in Process-Aware Information Systems.
3. Desel, J., Esparza, J.: Free Choice Petri nets. Volume 40 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (1995)
4. Schmidt, K.: Narrowing Petri net state spaces using the state equation. Fundam. Inform. **47**(3-4) (2001) 325–335
5. Schrijver, A.: Theory of Linear and Integer Programming. Wiley-Interscience series in discrete mathematics. John Wiley & Sons (1986)
6. Wolf, K.: On synthesizing behavior that is aware of semantical constraints. In: Proceedings of AWPN 2008. Volume 380 of CEUR Workshop Proceedings., CEUR-WS.org (2008) 49–54
7. Lohmann, N., Massuthe, P., Wolf, K.: Behavioral constraints for services. In: BPM 2007. Volume 4714 of LNCS. (2007) 271–287
8. Stahl, C., Wolf, K.: Deciding service composition and substitutability using extended operating guidelines. Data Knowl. Eng. (2008) (Accepted).
9. Fu, X., Bultan, T., Su, J.: Analysis of interacting BPEL web services. In: WWW '04, ACM (2004) 621–630
10. Schlingloff, B.H., Martens, A., Schmidt, K.: Modeling and model checking web services. Electr. Notes Theor. Comput. Sci. **126** (2005) 3–26
11. Mateescu, R., Poizat, P., Salaün, G.: Adaptation of service protocols using process algebra and on-the-fly reduction techniques. In: ICSOC. Volume 5364 of LNCS. (2008) 84–99
12. K. van Hee, Oanea, O., Sidorova, N., Voorhoeve, M.: Verifying generalized soundness for workflow nets. In: PSI. Volume 4378 of LNCS., Springer (2007) 235–247
13. Verbeek, H.M.W., van der Aalst, W.M.P.: Woflan 2.0: A Petri-net-based workflow diagnosis tool. In: ATPN 2000. Volume 1825 of LNCS., Springer (2000) 475–484