Checking service compatibility via resource conformance

Ivan Romanov
Software Engineering School
National Research University Higher School of Economics
33/5 Kirpichnaya, st. Moscow, Russian Federation
Email: romanov.ekb@gmail.com

Scientific Advisor: Prof. Irina A. Lomazova
Software Engineering School
National Research University Higher School of Economics
33/5 Kirpichnaya, st. Moscow, Russian Federation
Email: ilomazova@hse.ru

***Abstract* — In this work we consider modeling of services with workflow modules, which are a subclass of Petri nets. Service compatibility problem is studied. Quasi-regular expressions are used for checking service compatibility via resource conformance.**

Keywords – service composition; service compatibility; workflow net; workflow module

I. INTRODUCTION

Service-Oriented Computing [1] is an emerging computing paradigm that supports the modular design of (software) systems. Complex systems are designed by composing less complex systems, called *services*.

A service consists of a *control structure* describing its behavior and of an *interface* to communicate asynchronously with other services. Component interaction necessitates a mechanism of communication between services. An interface is a set of (input and output) channels. In order that two services can interact with each other, an input channel of the one service has to be connected with an output channel of the other service

There are a lot of algorithms for checking services compatibility. They allow, for example, to check deadlock freedom property, termination property, and others. Very often there methods have high computational complexity, affecting the speed of their implementation. It is assumed, that for service execution resources consumed from external environment are needed. In this paper a property of service compatibility via resource conformance is checked in the initial stage, it may help to avoid further verification because of evidently incompatible services.

In this work we consider modelling of services with workflow modules, which is a special subclass of Petri nets. Service compatibility problem is studied. Quasi-regular expressions are used for service compatibility via resource conformance analysis.

II. FORMAL MODEL FOR SERVICES

While analyzing and verifying service compatibility, let us abstract from underlying technologies and implementations and consider formal model based on Petri nets.

Petri nets is a popular formalism for modeling and analysis of distributed systems. A Petri net $N = (P;\ T;\ F)$ consists of a set of transitions T, a set of places P, and a flow relation $F \subseteq (P \times T) \cup (T \times P)$. Graphically, a place is represented by a circle, a transition by a box, and the flow relation by directed arcs between them. Whilst transitions represent dynamic elements, for example, an activity of a service places represent static elements for example, a condition to perform an activity of a service. A state of a Petri net is represented by a marking, which is a distribution of tokens over the places. Graphically, a token is depicted by a black dot.

Workflow net (WF net) is a special Petri net. This formalism is used to model workflow systems [3]. A workflow net has one initial and one final place, and every place or transition in it is on a directed path from the initial to the final place.

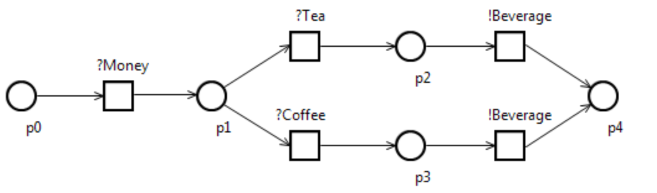

Figure 1. Example of workflow net. Model of a vending machine that sells, either a cup of tea, or coffee.

Modeling workflow [5] consists of modeling case management with the help of *parallelism*, *sequential routing*, *conditional routing* and *iteration*. To express it explicitly building blocks such as the AND-split, AND-join, OR-split and OR-join can be used. The AND-split and the AND-join are used for parallel routing. The OR-split and the OR-join are used for conditional routing. When we model an OR-split in terms of a Petri net, the OR-split corresponds to a number of transitions sharing the same set of input places. Other constructs also can be easily expressed in Petri net formalism.

To guarantee, that we get 'good' workflows, we are to balance AND/OR-splits and AND/OR-joins. Clearly, two parallel flows initiated by an AND-split, should not be joined by an OR-join. Two alternative flows created via an OR-split, should not be synchronized by an AND-join. When we follow these rules we obtain *structured WF nets* (see [3] for more details). It is shown there, that structured WF nets are sound by construction.

A stateful service defines an internal process (i.e. activities building its internal structure), and an interface to communicate with other services. To model services we will use a special subclass of Petri nets called workflow module.

A Petri net $M = (P; T; F)$ is called workflow module [2],[4] if the following conditions hold:

1) The set of places is divided into three disjoint sets: internal places P^N, input places P^I and output places P^O.

2) The flow relation is divided into internal flow $F^N \subseteq (P^N \times T) \cup (T \times P^N)$ and communication flow $F^C \subseteq (P^I \times T) \cup (T \times P^O)$.

3) The net $PM = (P^N; T; F^N)$ is a workflow net.

4) No transition is connected both to an input place and an output place.

Within a workflow module M, the workflow net PM is called the internal process of M and the tuple $\mathcal{I}(M) = (P^I; P^O)$ is called its interface.

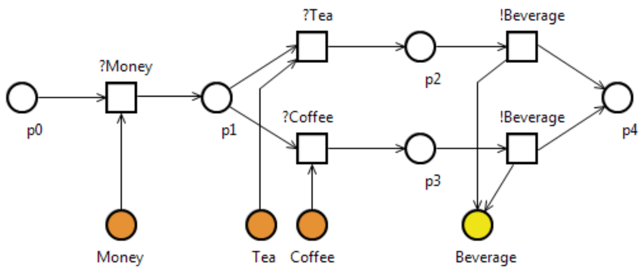

Figure 2. Example of a workflow module."Money","Tea","Coffee" are input places, "Beverage" is output place.

Denote structured workflow module (SWFM) as a workflow module, where the net $PM = (P^N; T; F^N)$ is a structured workflow net.

In this paper we consider the class of services, which can be modeled by SWFM.

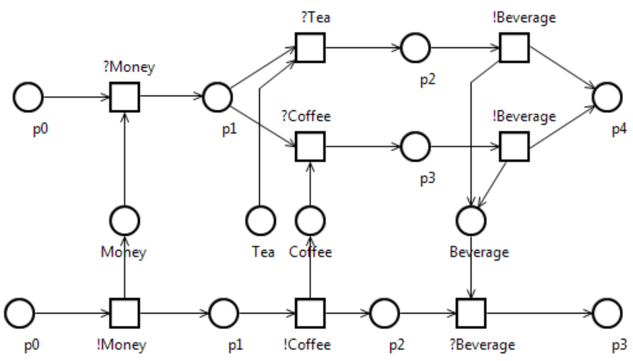

Figure 3. Example of service composition

Figure 3 illustrates an example of service composition. It shows situation when customer wants to buy coffee.

III. Service Compatibility via Resource Conformance

Let S be a finite set. A multiset m over a set S is a mapping m: S $\rightarrow$ $\mathbb{N}$, i.e. a multiset may contain several copies of the same element. By $\mathcal{M}(S)$ we denote the set of all finite multisets over S.

Let N is structured workflow module with two disjoint sets P^I of input places and P^O of output places. Consider some run δ, states sequence from initial to final state, of the N. For each run pair of input and output resources $(R_I;R_O)$,where $R_I \subseteq \mathcal{M}(P^I)$ and $R_O \subseteq \mathcal{M}(P^O)$ are defined. Denote $\rho(\delta) = (R_I;R_O)$ as resource of N [2].

For the considered class of services, let us denote quasi-regular expression defining a set of all possible resources for a service. For such type of resources, inner workflow net is structured, therefore, it becomes possible to identify the expression recursively by its structure:

1) Expression takes the form $R_I \circ R_O$ for atomic net where transition consumes R_I and produces R_O resources. Thus, rewrite this expression as

$$?r_1 \circ ?r_2 \circ \dots \circ ?r_k \circ !r_{k+1} \circ !r_{k+2} \circ \dots \circ !r_{k+m}$$

where $r_1, \dots, r_k$ – input resources; $r_{k+1}, \dots, r_{k+m}$ – output resources; If the transition does not produce/consume resources it is denoted as ε .

2) For sequential routing $e_1 \circ e_2$

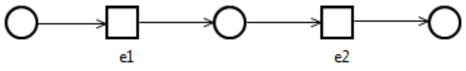

Figure 4. Sequential routing.

3) For parallel routing $e_1 \circ e_2$

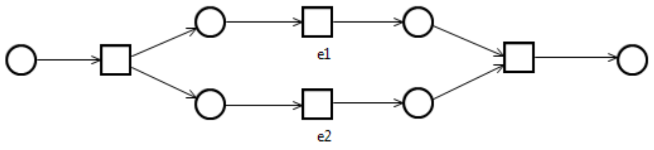

Figure 5. Parallel routing.

4) For selective routing $e_1 + e_2$

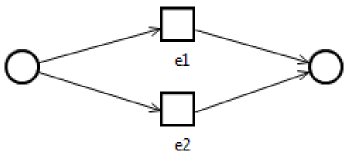

Figure 6. Selective routing.

5) For iteration $e_1 + (e_2 \circ e_1)^*$

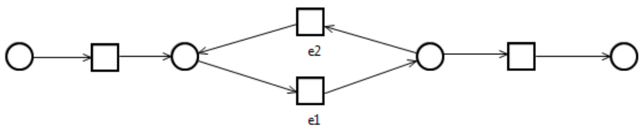

Figure 7. Iteration.

Proposition 1. Let expressions e1 and e2 define a set of resources for two structured workflow modules N_1 and N_2. Then after application of described operations deduced SWFM will possess the following resource sets: $e_1 \circ e_2$ for sequential, parallel routing, $e_1 + e_2$ for selective routing and $e_1 + (e_2 \circ e_1)^*$ for iteration.

Describe some properties of the algebra that we have:

1) $e \circ \varepsilon = e$
2) $\varepsilon^* = \varepsilon$
3) $e \circ e^* = e^+$
4) $(e^*)^* = e^*$
5) $e_1 \circ e_2 = e_2 \circ e_1$
6) $e_1 + e_2 = e_2 + e_1$
7) $e_1 \circ (e_2 + e_3) = e_1 \circ e_2 + e_1 \circ e_3$

For this class of quasi-regular expressions, as opposed to regular languages, a sequence of producing/consuming resources does not matter. That is why property 5 is true for quasi-regular expressions, while it is false for regular ones.

Using rules described, we may similarly compose a quasi-regular expression for the service investigated. To make representations more compact, we suppress interface conditions. Depicting a transition by a set of necessary resources is quite sufficient.

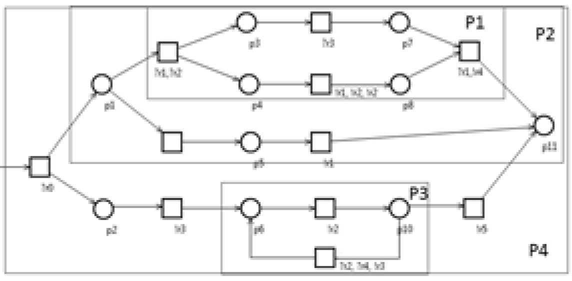

Figure 8. SWFN.

For example let us build quasi-regular expression for SWFN illustrated in Figure 8. We will consider resulted formula by parts P1, P2, P3, P4.

$$P1{:}\ (?r1 \circ ?r2 \circ (?r3 \circ !r1 \circ !r2 \circ !r2) \circ ?r1 \circ !r4)$$

$$P2{:}\ (!r1 + P1)$$

$$P3{:}\ !r2(?r2 \circ ?r4 \circ !r3 \circ !r2)$$

$$P4{:}\ ?r0 \circ (P2 \circ !\mathrm{r3} \circ \mathrm{P3} \circ !\mathrm{r5})$$

Expression for SWFN:

$$?r0 \circ ((!r1 + (?r1 \circ ?r2 \circ (?r3 \circ !r1 \circ !r2 \circ !r2) \circ ?r1 \circ !r4)) \circ !\mathrm{r3} \circ !r2(?r2 \circ ?r4 \circ !r3 \circ !r2) \circ !\mathrm{r5})$$

To check services equivalence and other properties, we need a unified representation for services. In order to achieve this, it is necessary to develop an algorithm for expressions reduction to normal form.

Two resources with equal representation, which is symmetric about '?' and '!' we denote as *complementary*. For example, $!\mathrm{r1} \circ ?\mathrm{r2} \circ !\mathrm{r3}$ and $?\mathrm{r1} \circ !\mathrm{r2} \circ ?\mathrm{r3}$ are complementary.

Two services we denote as *compatible via resource conformance* if their quasi-regular expressions contain two complementary resources.

Proposition 2. If service behaviors are compatible to each other, then two services are *compatible via resource conformance*.

Thus it is prerequisite condition for service compatibility.

IV. Conclusion

In this paper a service formal model is built and expression algebra for analyzing service compatibility via resource conformance, abstracting from underlying technologies and implementations is described. Detecting resource incompatibility in one of the first stages allows to avoid further verification consuming significant computing resources.

Further investigation in this field will be devoted to normal form construction for expressions, exploring new expression properties and compatibility proposition corroboration. Besides, a piece of software is planned to be developed. Its main functions will concern service model and service expression construction, and compatibility checking by the properties described.

References

[1] C. Stahl. Service Substitution – A Behavioral Approach Based on Petri Nets Dissertation, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II; Eindhoven University of Technology, December 2009.

[2] V. A. Bashkin and I. A. Lomazova. Resource equivalence in workflow nets Proc. CS&P'2006. Vol.1. Informatik-Bericht 206. Humboldt-Universitat zu Berlin. Berlin, 2006, pp. 80-91.

[3] W. van der Aalst and K. van Hee. Workflow Management: Models, Methods and Systems MIT Press,2002.

[4] Axel Martens. Analyzing Web Service based Business Processes. In Maura Cerioli, editor, International Conference on Fundamental Approaches to Software Engineering (FASE 2005), volume 3442 of Lecture Notes in Computer Science, Edinburgh, Scotland, April 2005. Springer. [36, 40, 216, 217]

[5] I. A. Lomazova. Interacting Workflow Nets for Workflow Process Re-Engineering. Fundamenta Informaticae, 2010. T.101.№1-2. C.59—70

[6] Peter Massuthe, Wolfgang Reisig, and Karsten Schmidt. An Operating Guideline Approach to the SOA. In 2nd South-East European Workshop on Formal Methods 2005 (SEEFM05), Ohrid, Republic of Macedonia, 2005.