Distributed management system for infocommunication networks based on TM Forum Framework

Valery P. Mochalov
mochalov.valery2015@yandex.ru

Natalya Yu. Bratchenko
nb20062@rambler.ru

Sergey V. Yakovlev
Yak0vlevSV@yandex.ru

Darya V. Gosteva
dvgus@yandex.ru

Department of Infocommunications, North-Caucasus University, Stavropol, Russian Federation

Abstract

The existing management systems for networks and communication services do not guarantee in full an efficient rendering of next-generation infocommunication services. The Open Digital Architecture (ODA) is expected to simplify dramatically and automate main business processes. A key principle of the ODA is using the component approach with the logic of microservices and Open APIs. The concept of Frameworx gives an exact description for the components of business processes in terms of their functions, associated information and other characteristics. The performance of a distributed operational management system depends on the quality of solving several tasks as follows: the distribution of program components among processor modules; the prioritization of business processes with parallel execution; the elimination of dead states and interlocks during execution; the reduction of system cost to integrate separate components of business processes. The deadlocks of parallel business processes are eliminated using an interpretation of the classical file sharing example with two processes and also the methodology of colored Petri nets, with implementation in CPN Tools. This paper develops a model of a distributed operational management system for next-generation communication networks that assesses the efficiency of operational activities for a communication company.

1 Introduction

Due to high competition in the field of telecommunications, network convergence processes and the need for new business models, it is important to develop methods of operational cost reduction and risk decrease in digital business management. Tele Management Forum (TM Forum) [TMF18], an international non-commercial association of telecommunication companies and their suppliers, is helping the member organizations to perform transformation into successful digital companies.

In: Marco Schaerf, Massimo Mecella, Drozdova Viktoria Igorevna, Kalmykov Igor Anatolievich (eds.): Proceedings of REMS 2018 – Russian Federation & Europe Multidisciplinary Symposium on Computer Science and ICT, Stavropol – Dombay, Russia, 15–20 October 2018, published at http://ceur-ws.org

The existing management systems for networks and communication services [SW11, LTK07, GI14] do not guarantee in full an efficient rendering of next-generation infocommunication services. TM Forum experts proposed the Open Digital Architecture (ODA) [ODA18], which is expected to replace traditional Operation Support Systems/Business Support Systems (OSSs/BSSs) as well as to simplify dramatically and automate main business processes (BPs). The ODA functionality (see Fig. 1) is intended to execute business processes without any human interference using advanced technologies, including artificial intelligence. A key principle of the ODA is using the component approach with the logic of microservices and Open APIs.

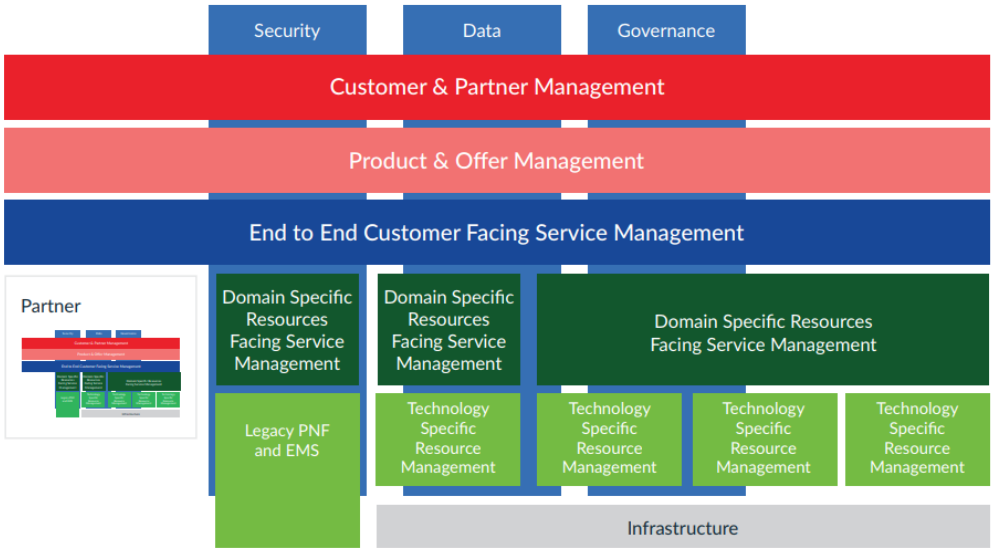

Figure 1: TM Forum Open Digital Architecture [ODA18]

2 Methods

2.1 Frameworx TM Forum

The ODA project (see Fig. 1) is implemented using the concept of Frameworx TM Forum, known earlier as New Generation Operations Systems and Software (NGOSS). This concept defines a modern standardization approach for the business processes of a communication company [Fra18]. Frameworx gives an exact description for the components of BPs in terms of their functions, associated information and other characteristics. Frameworx consists of the following frameworks (see Fig. 2).

1. The Business Process Framework (formally Enhanced Telecom Operations Map or eTOM) describes the structure of business processes of telecommunication companies.
2. The Information Framework (formally Shared Information/Data Model or SID) defines an approach to the description and usage of all data engaged in the business processes of a communication company.
3. The Application Framework (formally Telecom Applications Map or TAM) describes a typical structure of the Information Framework components for communication companies.
4. The Integration Framework contains a set of standards that support the integration and interoperability between applications defined in the Applications Framework, with a basic element in form of a standardized interface; a set of similar interfaces defines a service (API service).
5. The Business Metrics is a standardized model of business indicators that unites over a hundred of standard measurable indicators for assessing different activities of an infocommunications supplier.
6. The Best Practices includes practical recommendations and case studies based on the experience of using Frameworx models in different activities of telecommunication companies.

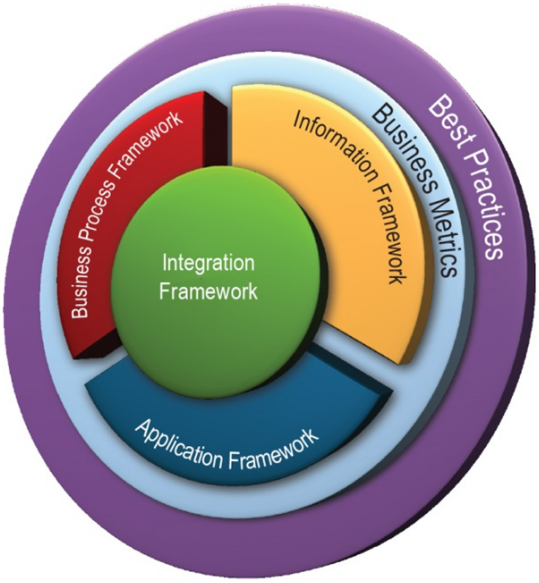

Figure 2: Structure of Frameworx TM Forum [Fra18]

The structural description of BPs in eTOM relies on the hierarchical decomposition principle. Consider the decomposition procedure for BP 1.4.6.3 (Correct & Resolve Service Problem); there are four decomposition levels here, as illustrated in Fig. 3.

The first decomposition level of eTOM contains eight large blocks:

- 1.1 Market/Sales Management Domain;
- 1.2 Product Management Domain;
- 1.3 Customer Management Domain;
- 1.4 Service Management Domain (shown in Fig. 3);
- 1.5 Resource Management Domain;
- 1.6 Engaged Party Domain;
- 1.7 Enterprise Domain;
- 1.8 Common Process Patterns Domain.

The second decomposition level separates out the groups of processes that repre-sent large stages of end-to-end business processes in eTOM. Block 1.4 Service Management Domain at level 2 is divided into eight groups (see Fig. 3) and includes, including, block 1.4.6 Problem Management Service.

The described levels are logical because the resulting specification does not yield a sequence of actions. The third and lower decomposition levels are physical, since their elements correspond to specific actions that can be combined in flows.

The processes of the third level decomposition can be used to construct ideal models without possible failures during their execution and other peculiarities. Block 1.4.6 Problem Management Service level 3 is divided into seven process groups (see Fig. 3) and includes, including, block 1.4.6.3 Correct & Resolve Service Problem.

Block 1.4.6.3 Correct & Resolve Service Problem on the decomposition level 4 is divided into seven processes (see Fig. 3):

- 1.4.6.3.1 Reassign / Reconfigure Failed Service;
- 1.4.6.3.2 Manage Service Restoration;

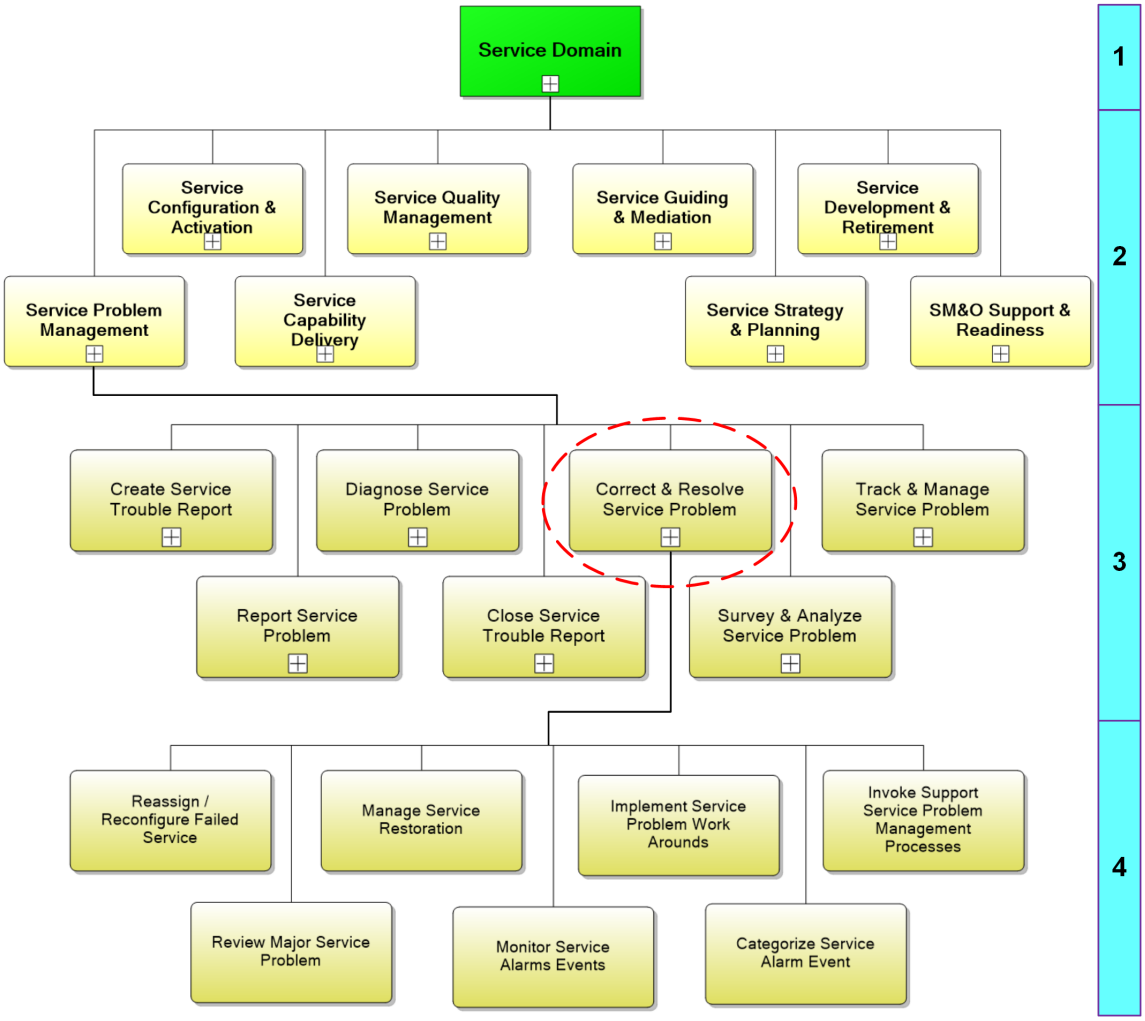

Figure 3: Decomposition of BP 1.4.6.3 (Correct & Resolve Service Problem) [Fra18]

- 1.4.6.3.3 Implement Service Problem Work Arounds;
- 1.4.6.3.4 Invoke Support Service Problem Management Processes;
- 1.4.6.3.5 Review Major Service Problem;
- 1.4.6.3.6 Monitor Service Alarms Events;
- 1.4.6.3.7 Categorize Service Alarm Event.

Using the specification elements obtained at the fourth level, it is possible to construct a detailed model of a business process for automated management (see Fig. 4).

The above approaches to management system design based on a set of modules implement the microservice architecture principles, which is used to create program systems composed of numerous multiple services interacting with each other. These modular components are intended for independent development, testing and deployment, which facilitates the creation of new services or deep update of the existing ones if necessary. Other advantages of such modules are cost reduction, speed increase and customer service improvement. The microservice architecture guarantees independent scalability, faster market entry for new services, and higher efficiency of management.

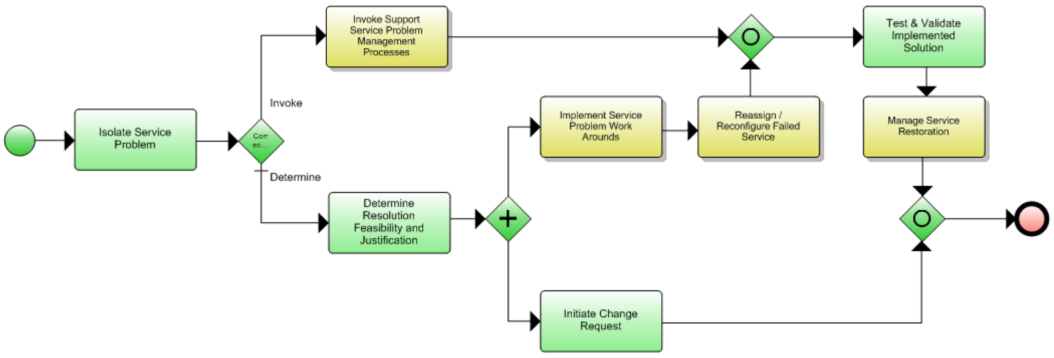

Figure 4: Sequence of steps for Correct & Resolve Service Problem [Fra18]

2.2 Model of distributed management system for next-generation networks

A distributed management system (DMS) for infocommunication networks can be defined as a set of modules or program components (PCs) executed by processor modules (PMs). Each PC implements a corresponding microservice or business component (BC) that is described within Frameworx. PMs can be implemented in form of physical devices or virtual resources. PCs interact with each other through the network adapters of PMs using the capabilities of real or virtual networks. Any service in this management system is implemented by assigning a certain set of PCs located on separate PMs for execution of a given BP.

Clearly, the performance of a distributed operational management system depends on the quality of solving several tasks as follows:

- the distribution of PCs among PMs (as a rule, the number of sets of PCs is much greater than the number of PMs), in this case, we assume that the real sequence of control transfers between PCs is approximated by the absorbing Markov chain;
- the prioritization of BPs with parallel execution as well as the elimination of dead states and interlocks during execution;
- the reduction of system cost to unite separate components of BPs.

A successful solution of these tasks yields an optimal DMS in terms of the criterion

$$\Phi = \min \sum_{k=1}^{L} P_k \times t_k\,(\Theta,\ S,\ Q),$$

with the following notations: t_k as the execution time of the kth BP;

Θ as a distribution method of PCs among PMs;

S as a service schedule of requests;

Q as an integration method for separate solutions of PMs;

L as the number of BPs served by the system;

finally, P_k as the execution priority of the kth BP.

To distribute PCs over PMs, consider n PCs $(f_1,\ \ldots,\ f_n)$ and also d PMs. Assume each two PCs, f_i and f_j, are exchanging joint requests with a known frequency $P\,(i,\ j)$. The average number of control transmissions between the i and j PCs can be obtained by taking measurements by the software monitor. We derive an expression for the average number of transitions between PCs in the process of service implementation.

Decompose the set of PCs into d groups $\Phi = \{F_1,\ F_2,\ \ldots,\ F_d\}$ so that

$$\bigcup_{i=1}^{n} f_i = \bigcup_{i=1}^{d} F_i \quad \textbf{and} \quad F_k \bigcap F_l = \emptyset, \quad k \neq l,$$

The frequency of conflicts on the kth PM is defined by

$$C_k = \sum_{i,\ j} p\,(i,\ j).$$

Then the total frequency of conflicts makes up

$$C = \sum_{k=1}^{d} C_k = \sum_{k=1}^{d} \sum_{i,\ j} p\,(i,\ j).$$

If PC_i moves between groups F_s and F_t, the optimality criterion has the variation

$$\Delta C = \sum_{i \in PM_i} (P\,(i,\ j) + P\,(j,\ i)) - \sum_{i \in PM_s} (P\,(i,\ j) + P\,(j,\ i))$$

All PC_i, $i = 1,\ \ldots, n$, are decomposed into d groups using a system of operators $R = \{R_{i\,t},\ i = 1,\ \ldots, n;\ t = 1,\ \ldots, d\}$ that formalizes the transition of PC_i to another class t, performing the operation $R_{i\,t}\Phi$. As a result, $\Delta_{i\,t}\,(\Phi) = C\,(R_{i\,t}\Phi) - C\,(\Phi)$, where $\Delta_{i\,t}\,(\Phi)$ is the increment of the optimal decomposition criterion.

Denote by $\Delta_{i\,t}^{i\,q}\,(\Phi)$ the increment of the values $\Delta_{i\,t}\,(\Phi)$ under transition to a new decomposition $R_{j\,q}\Phi$, where $R_{j\,q} \in R$, i.e.,

$$\Delta_{i\,t}^{i\,q}\,(\Phi) = \Delta_{i\,t}\,(R_{i\,q}\Phi) - \Delta_{i\,t}\,(\Phi)\,,\ i = 1,\ \ldots,\ n,\ q = 1,\ \ldots,\ d.$$

Consequently,

$$\Delta_{i\,t}\,(R_{i\,q}\Phi) = \Delta_{i\,t}\,(\Phi) + \Delta_{i\,t}^{j\,q}\,(\Phi)\,;\ \Delta_{i\,t}^{j\,q}\,(\Phi) = [P\,(i,\ j) + P\,(j,\ i)]\,(\delta_{t\,q} + \delta_{s\,u} + \delta_{s\,q} - \delta_{t\,u})\,,$$

where s and u are the indexes of groups for the PCs f_i and f_j (initial decomposition);

t and q are the indexes of their new groups (after transition); finally, $\delta_{t\,q} = \begin{cases} 1, & \textbf{if} \quad t = q, \\ 0, & \textbf{if} \quad t \neq q. \end{cases}$

For this criterion, the sequential improvement algorithm

$$\Delta_{i\,t}\,(R_{j\,q}\Phi) = \Delta_{i\,t}\,(\Phi) + \Delta_{i\,t}^{j\,q}\,(\Phi)$$

obtained under the assumption that the sequence of control transmissions between the PCs is described by the Markov chain, gives a rational distribution of PCs among PMs.

2.3 Elimination of dead states and interlocks

DMSs for infocommunication networks have complex operation algorithms with parallel processes. Note that some processes are interdependent because they share the same resources (e.g., hardware components, software tools, current information). Due to resource sharing, it is necessary to organize the interaction of parallel processes: their independent execution may cause errors, dead states or interlocks [Lee06, SL05, OAA09].

The dead states and interlocks occurring in parallel business processes are eliminated by scheduling. The efficiency of elimination is assessed using the indicator

$$T_{\min}\,(S) = \min\,\{t_i\,(S)\}\,,$$

where S means a schedule and $t_i\,(S)$ the service time of the ith request, subject to the constraint

$$t_{p+1}^{j} - t_p^i \geq Z_L^i - Z_L^j + t_{j\,L},\ L = \overline{1,\ k}.$$

The notations are the following: p as the index of the request of type i;
$(p+1)$ as the index of the request of type j;
L as the serial number of the PM;
Z_L^i and Z_L^j as the time cost to execute the requests of types i and j on the Lth PM;
finally, $t_{j\,L}$ as the time to execute the request of type j on the Lth PM.

Let $l_{ij} = \max\left\{Z_L^i - Z_L^j + t_{jL}\right\}$. Then the optimal service schedule of all requests is defined by the condition $t_{p+1}^j - t_p^i = l_{ij}$.

Describe the service procedure of all $n = \sum\limits_{k=1}^{r} n_k$ requests using a directed symmetric graph $(X,\ U)$, where $X = \{0,\ 1,\ \ldots,\ r\}$ and U is the set of arcs $(i,\ j)$, $0 \le i,\ j \le r$, each associated with the value l_{ij}. In this case, the optimal service schedule of all requests is the smallest loop in the graph that passes n_k times through each vertex.

If x_{ij} is the number of arcs in the desired loop, then

$$\sum_{i,\ j=0}^{r} l_{ij}x_{ij} \to \min;\ \sum_{j=0}^{r} x_{ij} = \sum_{j=0}^{r} x_{ji} = n_i;\ i = \overline{0,\ r};\ x_{ij} \ge 0;\ i,\ j = \overline{0,\ r}.$$

The deadlocks of parallel business processes are eliminated using an interpretation of the classical file sharing example with two processes [BS88, Piy86] and also the methodology of colored Petri nets [Jen13], with implementation in CPN Tools [JKW07, CPN18].

Consider the parallel implemented BPs, which consist of a sequence of operations $t_1,\ \ldots,\ t_k$ performed by PCs so that each operation corresponds to a transition in a Petri net (see Fig. 5). The user interface of the CPN Tools environment determines the Petri net marking. Contained in certain positions, the markers are highlighted in green, indicating their number, color and delay time value (for example, 1‘1 @ 0). Transitions are also highlighted in green, the triggering of which are currently allowed.

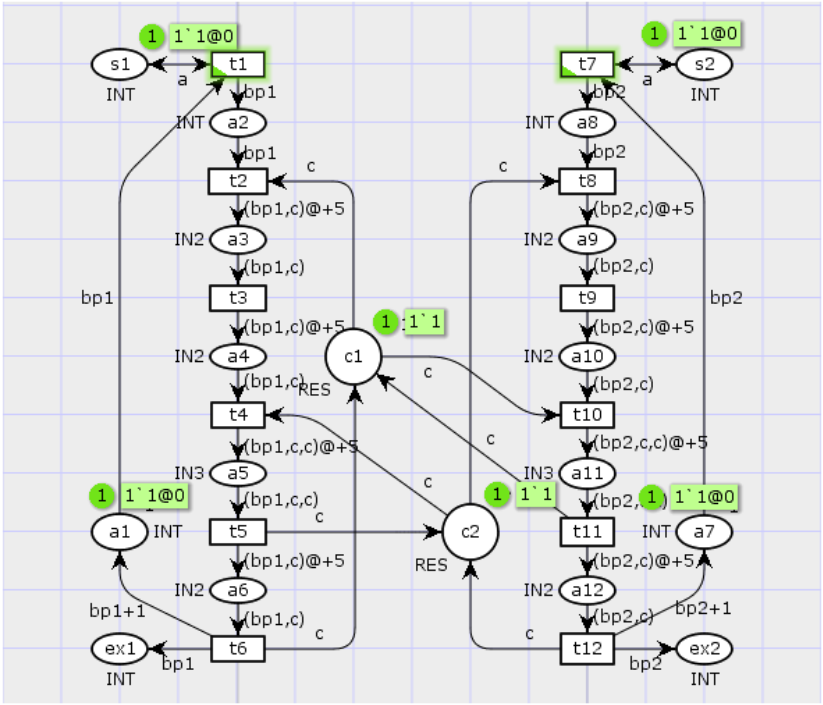

Figure 5: Initial state of execution model for two parallel BPs

A certain number of asynchronous parallel processes are competing for the right of resource use (*RES*). When a process holds a resource, the sequence of its operations is performed and the resource is considered to be busy. Since the same resource can be required for several processes, there may exist dead states and interlocks for them, as shown in Fig. 6. This condition occurs in the model under consideration after a certain number of allowed transitions are triggered (see Fig. 6). We see that at the moment there are no transitions in the network, the operation of which is allowed.

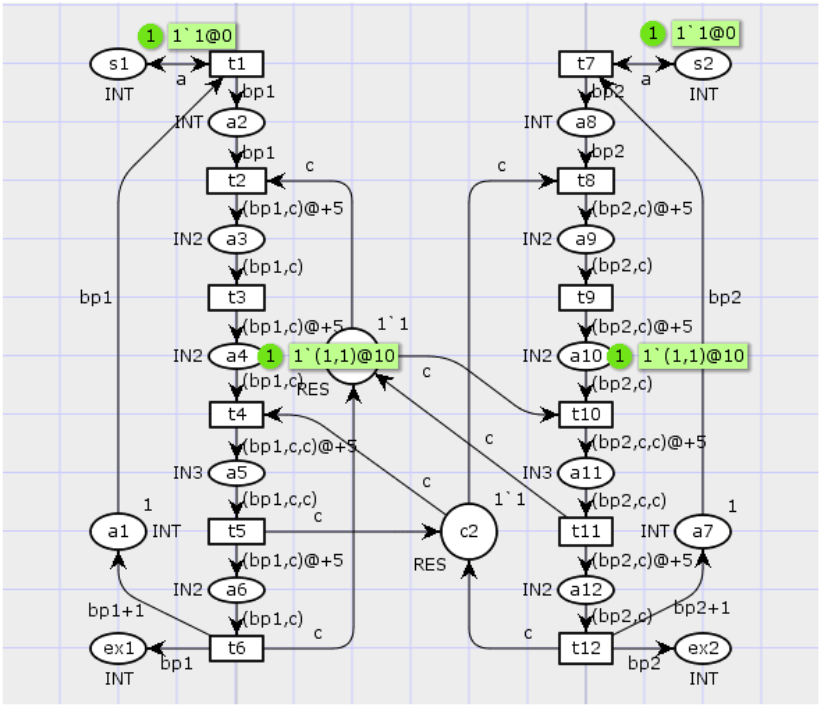

Figure 6: Execution model for two parallel BPs in interlock state

Our analysis of the model will employ the reachable label graph of the Petri net, see Fig. 7. The states of a Petri net from which all paths of the reachable label graph lead to a dead state (in our case, S_{33}) are called pre-dead states (in our case, S_{22}, S_{23}, and S_{32}). A set composed of the dead and pre-dead states is called the set of forbidden states [BS88]. Obviously, for a faster execution of processes, it is necessary to eliminate all conflicts using the available system parallelism as much as possible. Consider some ways to eliminate process interlocks and dead states.

The first approach is to use a well-timed forced locking of processes [BS88]. To this effect, define the minimal number of processes to be locked and also the minimal number of states to preserve this interlock. Then transform the reachable label graph by removing all the edges that connect dangerous and forbidden states. This procedure yields a graph containing the safe states only.

In our example, the state S_{11} is the root of two dangerous segments, $S_{21} - S_{31}$ and $S_{12} - S_{13}$ (see Fig. 8). The well-timed forced locking of undesired processes can be implemented by introducing an input position c_b for the transitions t_2 and t_8 in the Petri net (see Fig. 9). In this case, following the activation of the transition t_2 (t_8) and further evolvement of the process bp_1 (bp_2, respectively), the label is removed from the position c_b and the process bp_2 (bp_1, respectively) is locked.

In accordance with the transformed reachable label graph (see Fig. 8), the state S_{31} (S_{13}) is the last state of the chosen dangerous segment. So the lock can be lifted after the activation of the transition t_4 (t_{10}, respectively). To this end, the position c_b must be the output position for the transitions t_4 and t_{10}, as illustrated in Fig. 9.

The transformed reachable label graph (see Fig. 8) contains the position c_b in all states except for the ones corresponding to dangerous segments. The states S_{12}, S_{13}, S_{21}, and S_{31}, which are dangerous in the original graph, become safe lock states while the state S_{11} becomes the conflict state. In addition, the forbidden states S_{22}, S_{32}, S_{23}, and S_{33} turn out to be unreachable in the transformed graph. Simulation results for the transformed Petri net during 100 steps testify to the efficiency of this well-timed forced locking procedure, see Fig. 10.

The second approach to eliminate the dead states and interlocks of processes is to allocate additional resources required for their simultaneous execution, as shown in Fig. 11. In this case, the positions c_1 and c_2 have single labels, which corresponds to two units of homogeneous resource.

Next, the third approach to eliminate the dead states and interlocks of processes is to capture simultaneously all the resources required for a process (see Fig. 12). For lifting the interlock of the first process, a position c_{21}

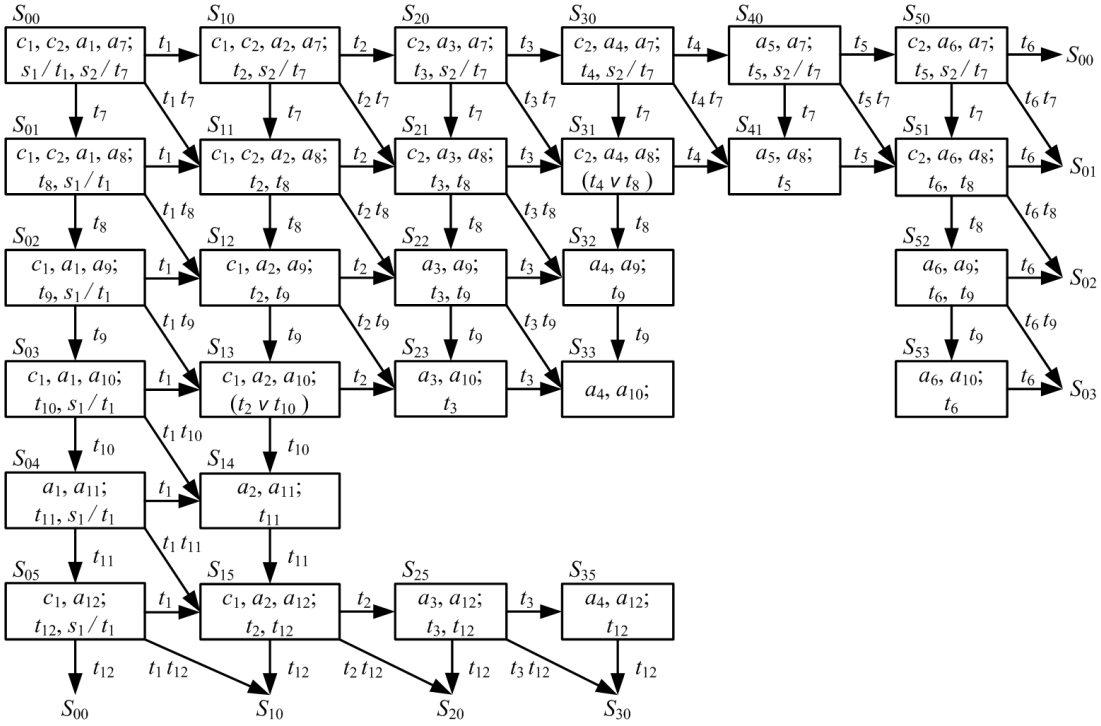

Figure 7: Reachable label graph of original Petri net

is added to the original network that holds the second resource.

The last (fourth) approach to eliminate the dead states and interlocks is to arrange the capture of resources. Serial numbers are assigned for all types of resources and a capture discipline is defined for all processes. In the transformed model, this approach is implemented by reassigning serial numbers for resources and specifying choice rules for the transitions t_2, t_4, and t_8, t_{10} (see Fig. 13).

The developed models eliminate the interlocks of parallel processes. The methodology of colored Petri nets is used to analyze the complete state space of the model in order to improve the reliability of the computing system and also to satisfy requirements. The suggested methods and software solutions accelerate and simplify program development, being finely integrated into the standard software approaches and methods and ready to be applied in practice.

3 Conclusions

This paper has considered formal problems of distributed operational management and support system design for communication companies and main approaches to solve them. A recursive convergent algorithm to distribute program components for an associated microservice or business component using Frameworx and also a service schedule of requests with the minimal number of delayed requests and colored Petri nets have been presented. All these add up to an efficient design of distributed operational management systems for future-generation telecommunication networks.

References

[TMF18] *TM Forum Homepage*. http://www.tmforum.org/. last accessed May 3, 2018.

[SW11] J. Simoes, S. Wahle. The future of services in next generation networks. *IEEE Potentials*, 30(1):24–29, February 2011. doi: 10.1109/MPOT.2010.939761.

[LTK07] T. Lechler, B. J. Taylor, B. Klingenberg. The Telecommunications Carriers' Dilemma: Innovation vs. Network Operation. *Management of Engineering and Technology, Portland International Center*, 2940–2947, August 2007. doi: 10.1109/PICMET.2007.4349638.

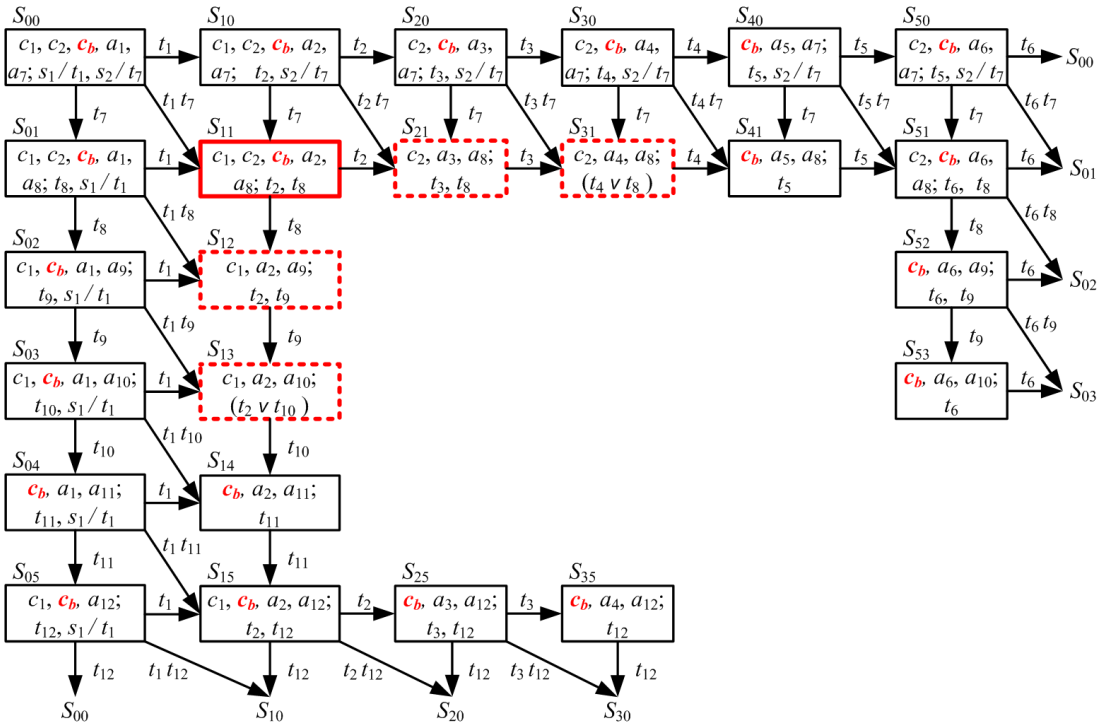

Figure 8: Transformed reachable label graph of Petri net

[GI14] V. V. Gluhov, I. V. Ilin. Project portfolio structure in a telecommunications company. *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, 509–518, August 2014. doi: 10.1007/978-3-319-10353-2_46.

[ODA18] *Open Digital Architecture.* http://www.tmforum.org/resources/whitepapers/open-digital-architecture/. last accessed May 3, 2018.

[Fra18] *Frameworx TM Forum.* https://www.tmforum.org/tm-forum-frameworx-2/. last accessed May 3, 2018.

[Lee06] E. A. Lee. The problem with threads. *Computer*, 39(5):33–42, May 2006. doi: 10.1109/MC.2006.180.

[SL05] H. Sutter, J. Larus. Software and the concurrency revolution. *Queue*, 3(7):54–62, September 2005. doi: 10.1145/1095408.1095421.

[OAA09] M. Olszewski, J. Ansel, S. Amarasinghe. Kendo: efficient deterministic multithreading in software. *ACM Sigplan Notices*, 44(3):97–108, March 2009. doi: 10.1145/1508284.1508256.

[BS88] L. Bic, A. C. Shaw. *The logical design of operating systems.*. 2nd ed. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1988.

[Piy86] E. I. Piyl. Ustranenie vzaimnoi blokirovki parallel'nykh protsessov pri statisticheskom raspredelenii resursov [Deadlock elimination for parallel processes with statistical allocation of resources]. *Network protocols and management in distributed computing systems: a compilation.* The USSR Academy of Sciences. Moscow, Nauka, 116–125, 1986. (in Russian).

[Jen13] K. Jensen. *Coloured Petri nets: basic concepts, analysis methods and practical use. Vol. 1.* Springer Science & Business Media, 2013.

[JKW07] K. Jensen, L. M. Kristensen, L. Wells. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3–4):213–254, March 2007.

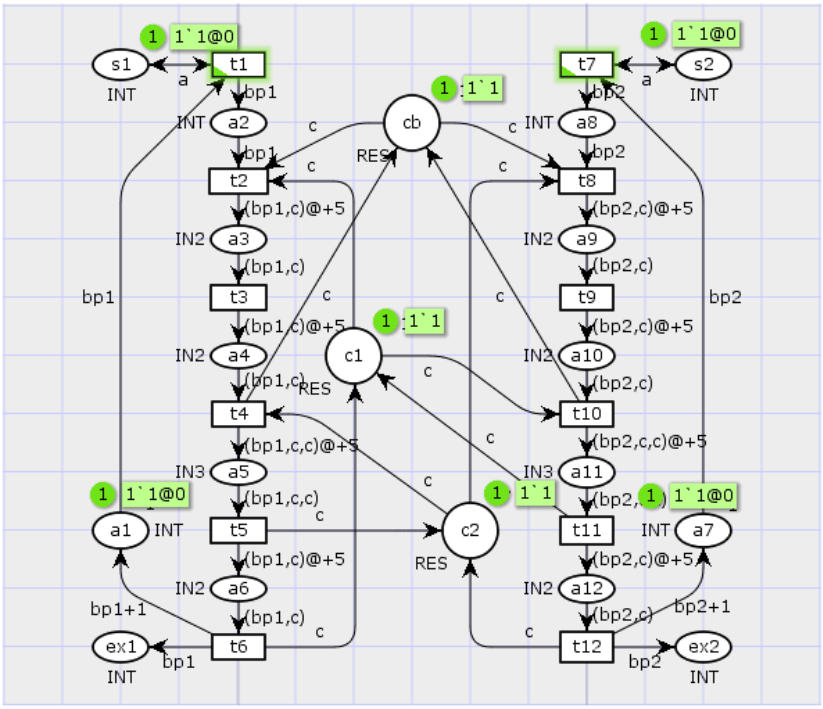

Figure 9: Initial state of execution model for two parallel BPs with well-timed forced locking

[CPN18] *CPN Tools Homepage*. http://cpntools.org/. last accessed May 3, 2018.

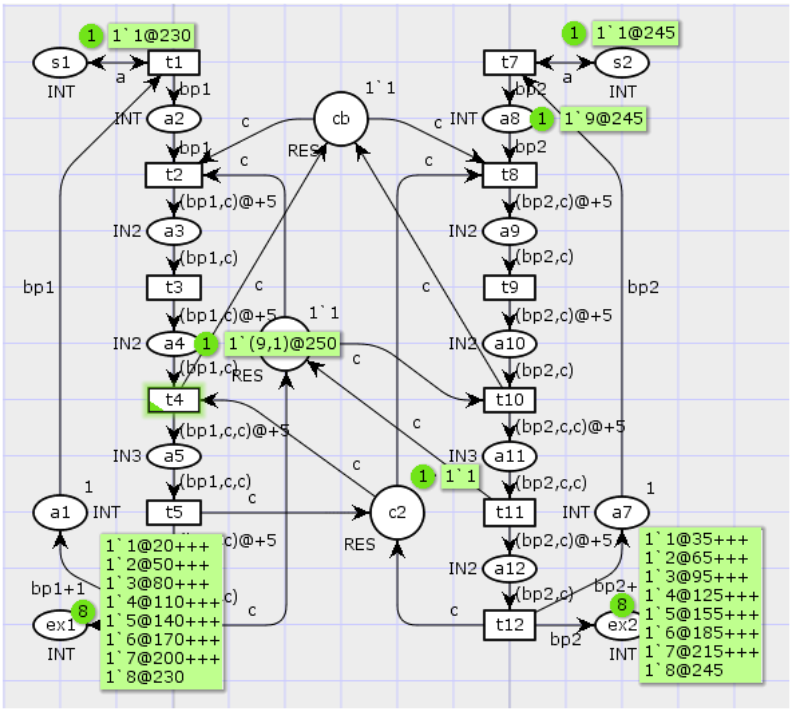

Figure 10: Transformed execution model for two parallel BPs: simulation results

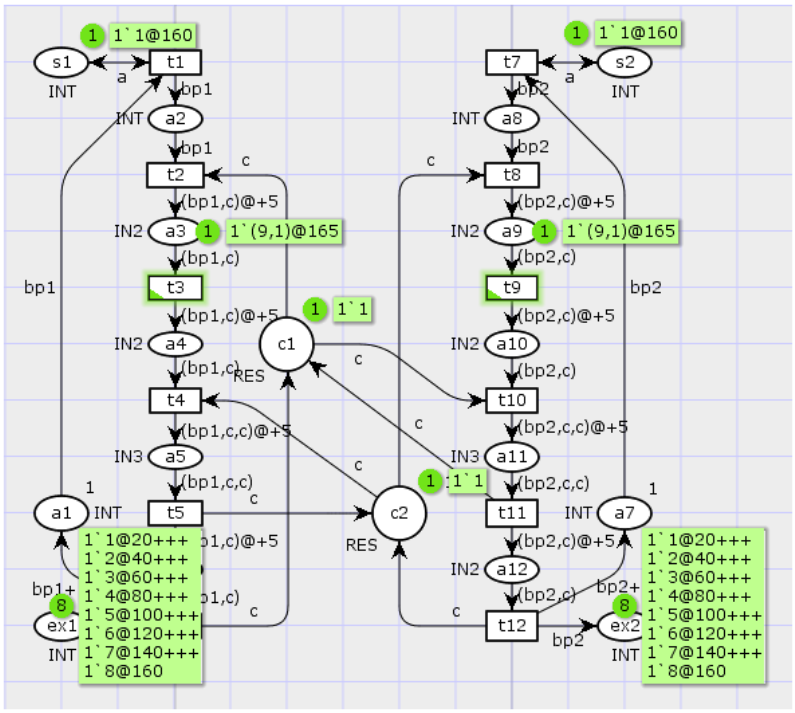

Figure 11: Execution model for two parallel BPs with allocation of additional resources: simulation results

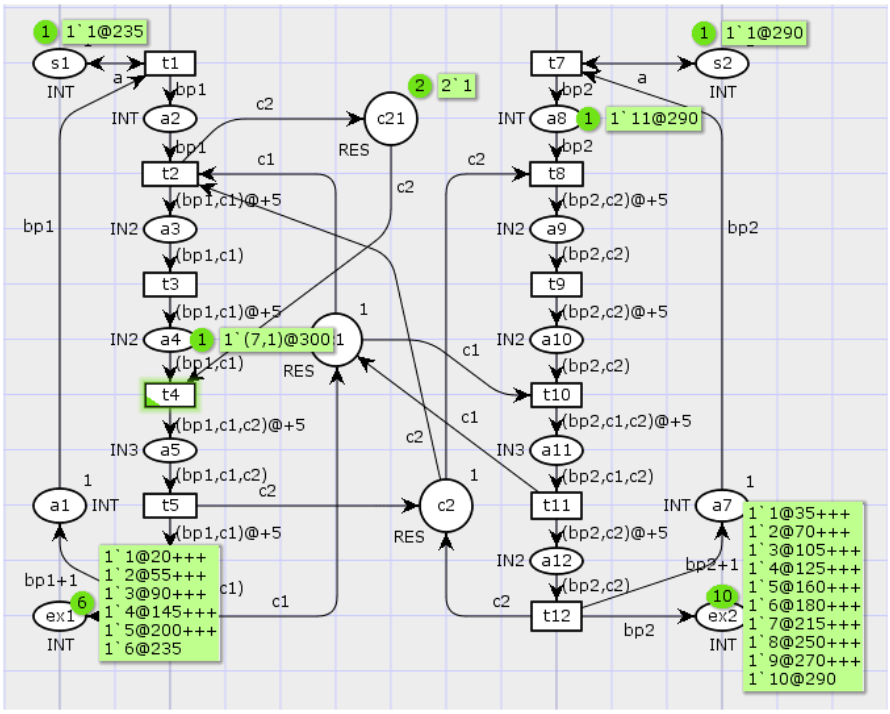

Figure 12: Execution model for two parallel BPs with monopolistic capture of resources: simulation results

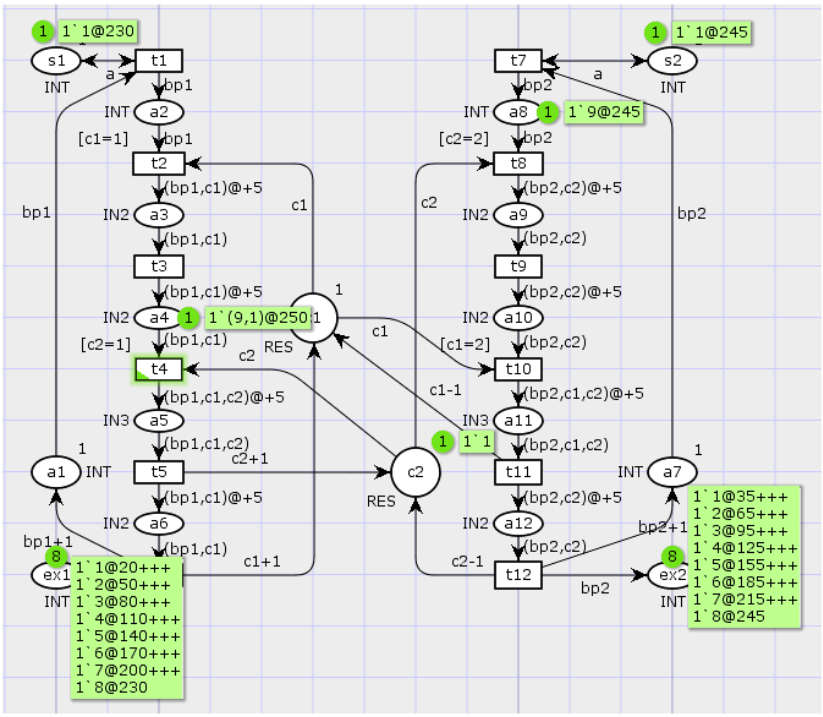

Figure 13: Execution model for two parallel BPs with arranged capture of resources: simulation results