

DOI: <https://doi.org/10.15276/aait.05.2022.14>

UDC 004.75.02

Method for developing and modelling composite web-services

Aleksandr P. Gozhyj¹⁾ORCID: <https://orcid.org/0000-0002-3517-580X>; alex.gozhyj@gmail.com. Scopus Author ID: 57198358626**Irina A. Kalinina¹⁾**ORCID: <https://orcid.org/0000-0001-8359-2045>; irina.kalinina1612@gmail.com. Scopus Author ID: 57198354193**Victor A. Gozhyj²⁾**ORCID: <https://orcid.org/0000-0002-5341-0973>; gozhyi.v@gmail.com. Scopus Author ID: 57198357517¹⁾ Petro Mohyla Black Sea National University, 10, 68 Desantnykiv Str. Mykolaiv, 54000, Ukraine²⁾ IT company “Template Monster”, 99A, Chkalov Str. Mykolaiv, 54056, Ukraine

ABSTRACT

The article describes a method for developing and modelling composite web-services. Web-service composition is used to derive new functionality from the interaction of existing web-services. Composite web-services are built in several stages: specifications (determining the type of service); development of the structure of the service based on the algebra of services; service composition modelling; selecting a service variant and generating a service. The main elements of the proposed approach are the algebra of services and web-services interaction models (basic and composite). The above approach formally presents the consideration of the main aspects in solving problems related to the construction of effective composite web-services and the selection of mathematical models, namely: description of web-services, determining the structure of web-services, taking into account the dynamics of information changes, taking into account the main uncertainties in building the structure web-services. Based on the approach, a method for constructing composite web-services has been developed. The method is based on an algebraic description of a web-service based on a specialized algebra of services. The method consists of the following steps: description of the task of developing a web-service; development of the structure of a web service based on the algebra of services; formal description of the structure and clarification of service operations; building models of functioning and interaction of service components; building simulation models of the composite service; defining the final structure of the composite service; web-service implementation. The main tool for describing composite services is the algebra of services, and the model building tool is coloured Petri nets implemented using CPN Tools. As an example of the application of the developed method, the construction of a pharmaceutical service is described, in which the modelling of the interaction of web-services based on Petri nets is used. The process of building a composite model of a web-service is considered. The process begins with building a pharmaceutical web-service model structure based on the service algebra. Based on the service algebra, operators were developed to implement composite services. The structure of the model is developed, which is presented in the form of a formal description. The process of building pharmaceutical service models in the CPN Tools environment is described in detail.

Keywords: Composite web-service; service algebra; service structure; composite service model; coloured Petri nets

For citation: Gozhyj A. P., Kalinina I. A., Gozhyj V. A. “Method for developing and modelling composite web-services”. *Herald of Advanced Information Technology*. 2022; Vol. 5 No. 3: 185–197. DOI: <https://doi.org/10.15276/aait.05.2022.14>

INTRODUCTION

The main element of a service-oriented architecture (SOA) is a structure consisting of services, on the basis of which the development of applied distributed applications is carried out. Development is carried out in heterogeneous environments. Services are combined into groups according to composition rules. The main requirement for services created according to the rules of composition is the possibility of reusing their components in other services [1]. Web-service composition is used to derive new functionality by combining existing web-services. Service composition extends aggregation by searching for

services when solving the problem of building new services given a high-level description of the problem. If parameter transformations are required for service composition, then the user must select these services manually and implement the composition. On the other hand, by specifying the semantic mark up of services, the information necessary for the selection and composition of services is available through a system of semantic descriptions of requests and service capabilities that provide automatic composition.

Web-services in general can be divided into two classes – simple and complex, or *atomic* and *composite* [2]. An atomic web-service runs on a request that comes in the form of a message, performs

© Gozhyj A., Kalinina I., Gozhyj V., 2022

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/deed.uk>)

its task, and possibly generates one response per request. An atomic web-service is typically a stateless service, in contrast to composite services, which are defined as services composed of multiple atomic services and can use extended interaction with external services [4, 5], [7, 8].

A composite web service includes a data flow defined by a service execution plan from one step to the next, whose behaviour depends on the state of the information being processed by the service. For example, the process of searching for information may contain the order in which the search for the necessary resources, querying them by certain parameters, searching for information, etc. is carried out. This set of related and structured procedures that implement the required task can be defined as the web service *composition* process.

Although service composition results in more complex and more advanced services, this approach has a number of problems, such as security and quality of service. By leveraging multiple services from interacting with multiple providers in the composition process, the security issue increases along with the issue of optimal service management. In addition, there are service performance issues with automated composition.

A web-service is a component written in any language that can be deployed on any platform that has a standard XML-based interface. The behavior of web-services is basically a partially ordered set of operations that is specified at design time. This makes them suitable for display and simulation using Petri nets. The theory of Petri nets makes it possible to model systems of various types of complexity using their mathematical representation in the form of Petri nets [9]. Analyzing the results of modeling based on Petri nets, one can obtain important information about the structure and dynamics of the behavior of the simulated system. This information is used to analyze the system and to improve it further. When modeling web services, operations are modeled by transitions and service states are modeled by positions. Arrows between positions and transitions are used to indicate causal relationships [9, 10].

There are many tools that allow modeling complex processes and systems using the apparatus of Petri nets. But among them, CPN Tools is the most functionally developed [11]. CPN Tools is a modeling system that uses the formalism of colored Petri nets to describe various models of systems and processes. The level of functional and service capabilities allows using CPN Tools to model

various multi-level complex systems, including composite web-services.

ANALYSIS OF LITERATURE DATA AND STATEMENT OF THE PROBLEM

Complex composite services are currently in high demand. Composing existing web services to create new efficient services is a requirement in many business areas. Service composition extends the functionality of services as an information processing tool. The operations required to select and compose services are implemented through semantic descriptions of the requirements and capabilities of composite web-services.

The composition process is considered a complex process for many reasons. According to [3], the complexity of the composition of services increases for the following main reasons. First, the dramatic increase in the number of web-services available over the Internet results in an increase in available service repositories for search. Moreover, web-services continue to change and update, require dynamic composition at runtime, which requires high performance computing. The next reason is that when different providers develop web-services in different models, making the mapping between services further complicates the task of composition.

Methodologies for creating web-services can be divided into syntactic and semantic. In syntactic composition, only the syntax of the web-service description is used. Is considered to build a workflow of composition and service dependency management through service inputs and outputs [24]. There are two main approaches to syntactic composition. First, a centralized approach, in which an intermediary is responsible for bringing services together and matching dependencies between services. Secondly, the decentralized approach, in this approach, the composition is performed through the interaction of services through direct interaction.

Semantic composition takes into account the meaning and purpose of services. Machines can automatically select, create and execute web services to achieve a specified task in accordance with user constraints [5, 25], [26]. Semantic composition is achieved based on the semantic description of web services that support the creation of an ontology to define various aspects of the functioning of web services.

The work [12] considers an approach to automatic composition of services based on the technology of semantic agents. The approach allows you to combine services that are semantically similar and have common goals into a single service.

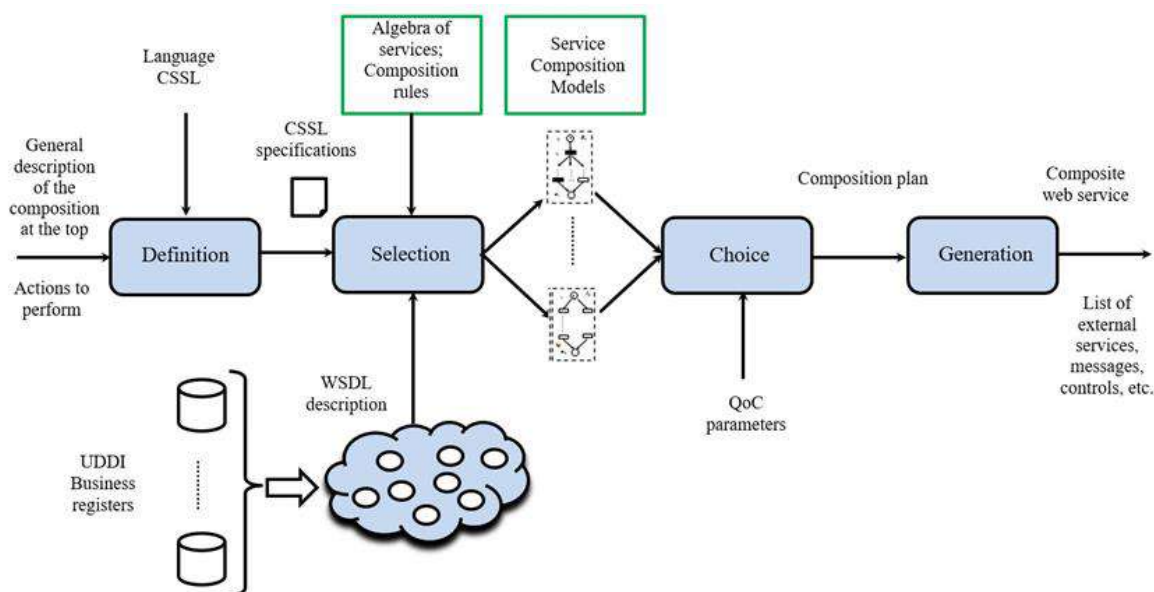


Fig. 1. Stages of a composite web-service development project implementation

Source: compiled by the authors

AI scheduling algorithms are used to compose services automatically and dynamically. In [6], a test planner model is proposed in which the user sets a goal used by the AI scheduling algorithm to create a composition plan in goal-oriented service architecture. Plans for this approach are built entirely at the outset and then they are executed. However, the dynamic nature of the SOA environment can cause the composition created by this approach to fail. The work [13] proposes automatic composition of services based on methods and algorithms of intelligent scheduling. In [14, 15], it is proposed to use automatic schedulers united by a common goal as a tool for composing services.

The issues of modelling the structures of web services based on ontology's are considered in [16, 17], [18, 19]. Approaches to choosing the structure of composite services based on selection algorithms are considered in [20].

The papers [21, 22] consider approaches to designing composite web-services based on evolutionary methods and genetic algorithms. From the many components of services are composed chromosomes, from which services are "assembled" with the help of crossing and mutation operations. The quality of services is evaluated by the fitness function. In [23], it is proposed to evaluate the quality of composite web services using the method of analytical hierarchies.

The modelling method based on Petri nets is proposed as a tool for building and composing web-services in [27]. The authors propose an approach to the description of web services based on

an algebraic description. Formalism for building a model of services - coloured Petri nets. In [28, 29], [30, 31], Petri nets of different types were used to build various web services.

Petri nets are a powerful tool for creating composite service models, so it is of interest to use different types of coloured Petri nets to create web-service models.

PURPOSE AND TASKS OF RESEARCH

The purpose of the article is to study the features of the web-services composition process and develop a method for constructing and modelling composite web-services based on coloured Petri nets.

METHOD FOR DEVELOPING AND MODELLING COMPOSITE SERVICES

Composite web-services are built in several stages: specification development (determining the type of service); development of the structure of the service based on the algebra of services; service composition modelling; service option selection and service generation (Fig. 1). The main elements of the proposed approach are the algebra of services with layout rules and web services composition models.

The description of a web-service in terms of the composition of web-services can be represented as follows:

$$KS = \langle Din, Q, WS, OI, InWS, Dout, T \rangle,$$

where $Din = \{dini \mid i = \overline{1, N_{Din}}\}$ – is a set of input data; $Q = \{qi \mid i = \overline{1, N_Q}\}$ – many user requests; $WS = \{wsi \mid i = \overline{1, N_{WS}}\}$ – is a set of basic web-services, components of the compositional system; OI – is a set of basic operations for the interaction of web-services; $InWS = \{inwsi \mid i = \overline{1, N_{InWS}}\}$ – a set of rules for the interaction of web-services; $Dout = \{douti \mid i = \overline{1, N_{Dout}}\}$ – a set of output data; $T = \{ti \mid i = \overline{1, N_T}\}$ – a set of temporal indicators of the interaction of the elements of the compositional model.

When solving problems of the composition of web-services, various mathematical models are used.

To determine the type of mathematical model, it is necessary to solve the following tasks:

- selection of a set of web service models that are most effective for solving the problems of building models of the dynamic composition of web services under given aggregation conditions;
- choice of a set of operations from the list of basic or combined ones to create web-service models;
- combining models into a single system, to solve the problems of dynamic composition of web-services during development, for example, for the task of multi-criteria choice of the structure of a web-service;
- inter-model matching in a composite web-service.

To solve these problems, an approach has been developed based on web service composition models, based on an algebraic-logical description of the rules for the functioning of a web-service. The approach consists of the following steps.

Stage 1. Preliminary definition of the structure of the web-service WSx , which is matched to the set of feasible solutions $WSx_{S\beta}$, the mathematical model M_v , within which the structure of the web-service x is determined:

$$M_v = \langle WSx \in WSx_{S\beta}, f_j(x), j \in J_v \rangle.$$

Stage 2. Presentation of the problem of model selection of efficient service structures.

Formalization is carried out in the following form:

$$WStr^M = \langle M, \{r_i^{M(\alpha)}\}, \{r_i^{M(\beta)}\}, i \in N^M \rangle.$$

Stage 3. Distribution of indicators for assessing quality and efficiency.

Distribution with the help of models is represented in the form of a mathematical structure of choice:

$$WSt^X = \langle X^M, \{r_i^{X(\alpha)}\}, \{r_i^{X(\beta)}\}, i \in N^X \rangle.$$

Stage 4. Inter-model matching.

The combination of the set M into a single service according to the rule π is represented as the following choice structure:

$$WSt^\pi = \langle M, \pi, \{r_i^{\pi(\alpha)}\}, \{r_i^{\pi(\beta)}\}, i \in N^\pi \rangle.$$

Stage 5. Building efficient service models.

The problem of constructing efficient service models involves solving selection problems in the structures WSt^M , WSt^X , WSt^π . Formally, this task is presented in the form of the following structure:

$$WSt = \langle WSt^M, WSt^X, WSt^\pi, \{r_i^\beta\} \rangle,$$

where r_i^β denotes relations that limit the choice of effective solutions in the development of composite web-services.

When developing web-systems, it is necessary to take into account the mechanisms of interaction between web-services, which give an idea of the structure of the web-system and the links of web-services when processing information of various types. Dynamic service composition models reflect the functionality of the web-system and allow you to detail the mechanisms of their composition.

The above approach formally presents the consideration of the main aspects in solving problems related to the construction of effective composite web-services and the selection of mathematical models, namely: determining the set of web-services and determining the structure of the web-service, taking into account the dynamics of information changes, taking into account the main uncertainties in the construction web-service structures. The developed model of a composite web-service allows you to describe a web service and use it in the process of building other web-

services to save data about the structure of a web-service.

Based on this approach, a method for constructing composite web-services has been developed (Fig. 2). The method is based on an algebraic description of a web-service based on a specialized service algebra [31].

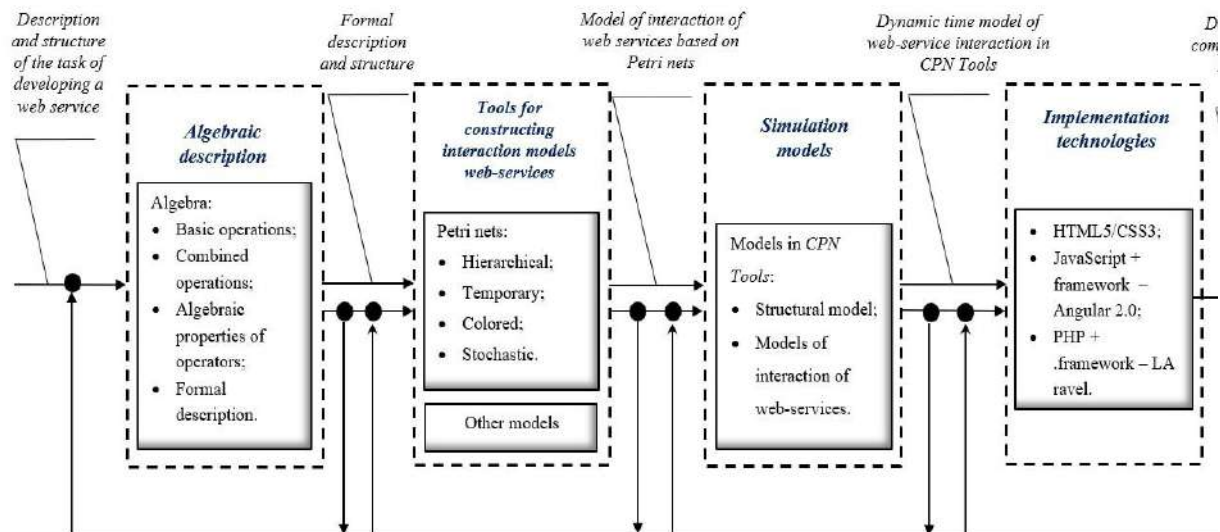


Fig. 2. Method for developing a composite web-service

Source: compiled by the authors

The method for developing and modeling composite web-services consists of the following steps:

- description of the task of developing a web-service;
- development of the structure of a web-service based on the algebra of services;
- formal description of the structure and clarification of service operations;
- building models of functioning and interaction of service components;
- building simulation models of a composite service (structural and interaction);
- determination of the final structure of the composite service;
- implementation of the web service.

The main tools for creating composite services are service algebra and colored Petri nets, implemented using CPN Tools. The service algebra consists of basic and combined operations. In algebra, the algebraic properties of operators are presented and described, and a toolkit is proposed for a formal description of the developed web-service [32]. At the second stage of the method, based on the formal description of the composite web-service, models of the functioning and interaction of the service components are built. Models are built on the basis of coloured Petri nets. At the next stage, simulation models of the structure of the composite web-service and models of the interaction of web-services in the composite service are built. Based on the simulation experiment, the

best structure of the composite web-service is determined and it is implemented using web development technologies.

BUILDING MODELS OF A COMPOSITE WEB SERVICE

For the joint use of basic and combined operations of the algebra of services, an example of building a composite (pharmaceutical) service is presented, in which modelling of the interaction of web services based on Petri nets is used [33].

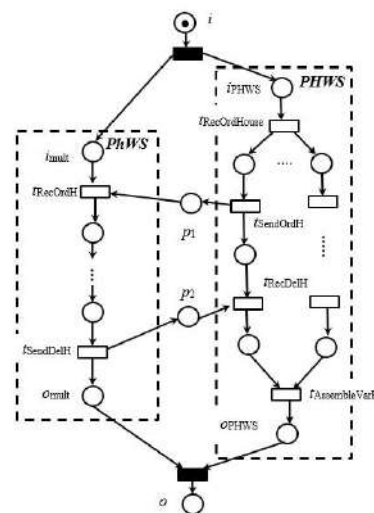


Fig. 3. Pharmaceutical web service structure

Source: compiled by the authors

Figure 3 shows the structure of a pharmaceutical web-service ($PhWS \parallel_c PHWS$), consisting of two basic services $PHWS$ (pharmaceutical service) and $PhWS$ (pharmacy services). The basic services are combined by a

parallel telecom operator, simulating the execution of a combined service, which activates the execution of two *PHWS* and *PhWS* services simultaneously with synchronization and information exchange.

After receiving a request from a client, *PHWS* launches an external request in parallel to obtain drug purchase options from the *PhWS* service. When performing this operation, *PHWS* receives *Ord_H* and sends *Del_H* accordingly. A set of elements for connecting services: $C_1 = \{(send_ord_H, rec_ord_H), (send_del_H, rec_del_H)\}$ and $C_2 = \{(send_ord_H, rec_ord_H), (send_del_H, rec_del_H)\}$. As soon as the options are received, the *PHWS* service aggregates the received information. Figure 4 shows that the *PhWS* composite service is used instead of the *PhWS* base service. Drug choices can be defined among available drugs at different pharmacies (eg, WS_1, WS_n) which are separate services of different pharmacy networks. Thus, replacing the service of one network of pharmacies according to the structure of dynamic selection allows you to choose the best service provider among several competing providers.

On the example of using the dynamic selection service, let's consider the interaction of services. Each service provider consists of two separate parts,

one part is designed to process a request for a service, and the second part is designed to perform the service itself. The two operations *SendReqServ* and *SelectServ* define the chosen selection strategy and the constructor of the composite service. The choice decision can be based on the automatic execution of services where the actions of the participants are automated.

Based on the algebra of services, operators were developed for the implementation of composite services, which are presented in Table 1. The proposed algebra checks the closure property. This ensures that every result of an operation on services is a service on which algebraic operators can be applied. Thus, it is possible to build more complex and combined services by aggregating and reusing existing services through declarative expressions of the service algebra.

Consider building a composite model of a pharmaceutical web-service using a colored Petri net. The process of building a composite web service model begins with building the structure of a pharmaceutical web-service model based on the service algebra [32].

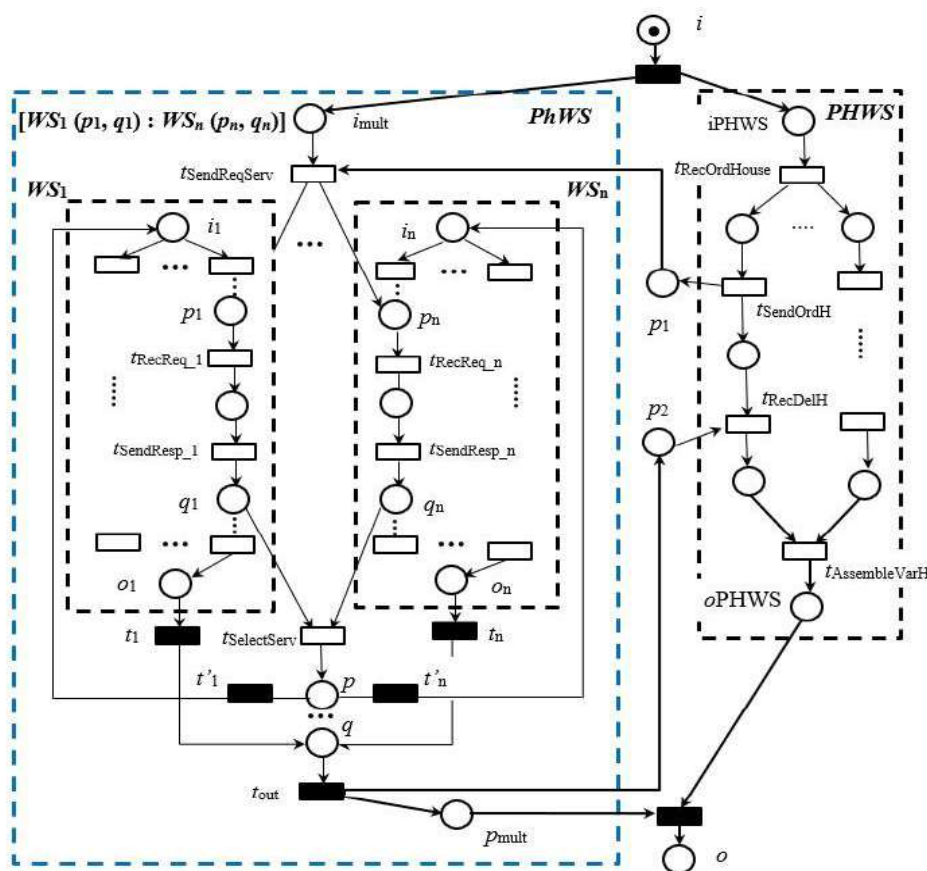


Fig. 4. Composite Pharmaceutical Service $[WS_1(p_1, q_1) : WS_n(p_n, q_n)] //c PHWS$

Source: compiled by the authors

Table 1. Use of the service algebra construct

Operator notation	Operator Description	Mathematical description	Graphical representation
Parallel operator with communication $WS_1 //c WS_2$	Parallel operator with communication: $WS_1 //c WS_2$ models the execution of a combined service that executes two services at the same time WS_3 and WS_2 with synchronization and information exchange	$WS_1 //c WS_2 = (NameS, Desc, Loc, URL, CS, SN)$, where: $-C$ – communication element $C = \{(t_\alpha, t_\beta) / (t_\alpha, t_\beta) \in T_3 \times T_2 \cup T_2 \times T_3\}$ $-NameWs$ — is the name of the new service, $-Desc$ — description of the new service, $-Loc$ — is the location of the new service, $-URL$ — is a new service call, $-CS = WS_3 \cup WS_2$, $-SN = (P, T, W, i, o, l)$ where: $\bullet P = P_3 \cup P_2 \cup \{i, o\} \cup \{p_i / (a_i, \beta_i) \in C\}$, $\bullet T = T_3 \cup T_2 \cup \{st_1, st_2\}$, $\bullet W = W_3 \cup W_2 \cup \{(i, st_1), (st_1, i_3), (st_1, i_2), (o_3, st_2), (o_2, st_2), (st_2, o)\} \cup \{(t_{a_i}, p_i), (p_i, t_{\beta_i}) / (t_{a_i}, t_{\beta_i}) \in C\}$, and $\bullet l = l_3 \cup l_2 \cup \{(st_1, \tau), (st_2, \tau)\}$.	
Dynamic Selection $[WS_1(p_1, q_1): WS_n(p_n, q_n)]$	The dynamic selection operator models the execution of a combined service that allows you to choose the best service provider among several competing providers. The dynamic selection construct allows you to choose the best service provider among existing ones using ranking criteria.	$[WS_1(p_1, q_1): WS_n(p_n, q_n)] = (NameS, Desc, Loc, URL, CS, SN)$, where: $-NameWs$ — is the name of the new service, $-Desc$ — description of the new service, $-Loc$ — is the location of the new service, $-URL$ — is a new service call, $-CS = \bigcup_{i=1}^n WS_i$ $-SN = (P, T, W, i, o, l)$ where $\bullet P = \bigcup_{i=1}^n P_i \cup \{i, o, p, q\}$, $\bullet T = \bigcup_{i=1}^n T_i \cup \{t_{SendReqServ}, t_{SelectServ}, t_o\} \cup \{t_i, t_i' \mid 1 < i < n\}$, $\bullet W = \bigcup_{i=1}^n W_i \cup \{(i, t_{SendReqServ}), (t_{SelectServ}, p), (q, t_o), (t_o, o)\} \cup \{(t_{SendReqServ}, p_i), (q_i, t_{SelectServ}), (p, t_i'), (t_i', i_i), (o_i, t_j), (t_j, q) \mid 1 < i < n\}$, and $\bullet l = \bigcup_{i=1}^n l_i \cup \{(t_{SendReqServ}, \tau), (t_{SelectServ}, \tau), (t_o, \tau)\} \cup \{(t_i, \tau), (t_i', \tau) \mid 1 < i < n\}$.	

Source: compiled by the authors



To build a composite web-service model in CPN Tools, three separate model sheets in CPN Tools are used. The first sheet shows a general model of interaction between top-level web services (Fig. 6), the second sheet shows a second-level network that models the process of generating requests (Fig. 7), and the third sheet shows a second-level network that models the process of interaction between web-services (Fig. 8) based on the composite pharmaceutical service model shown in Fig. 4.

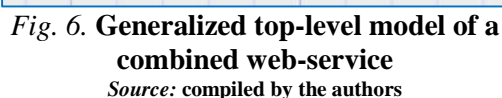


Fig. 7. Web-service request generator model
Source: compiled by the authors

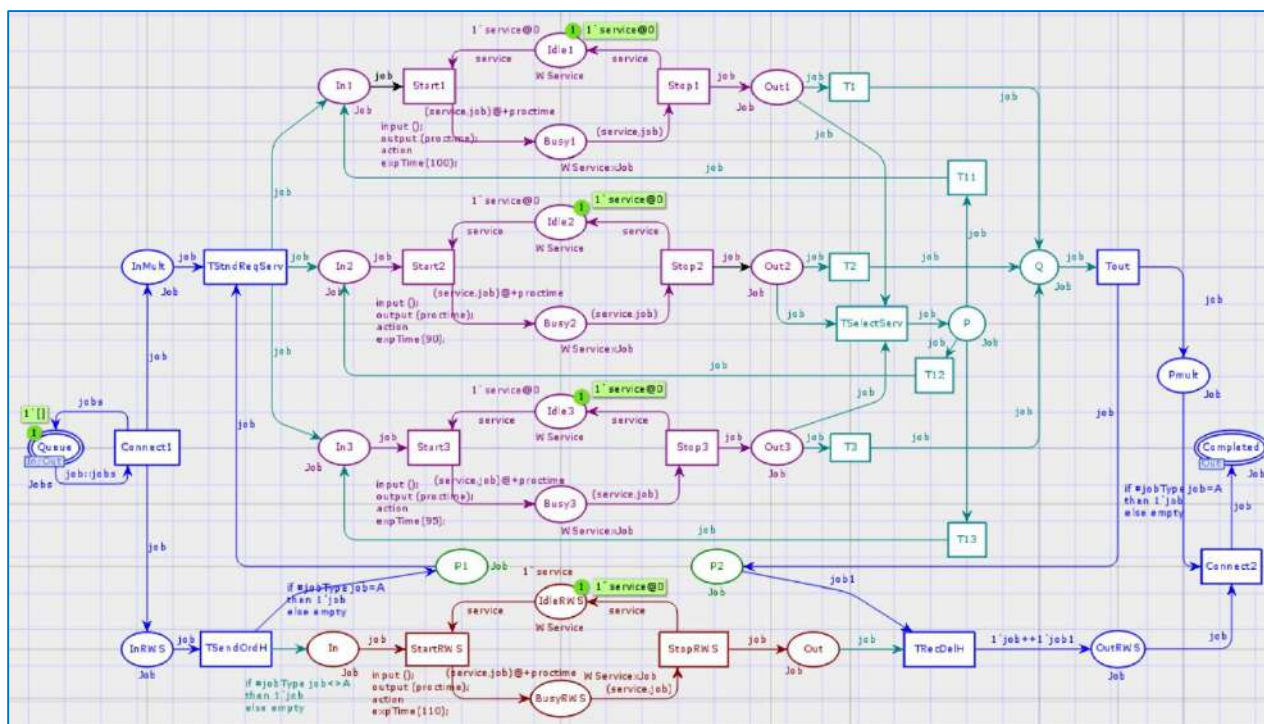


Fig. 8. Simulation model of the processing of requests by a combined web-service

Source: compiled by the authors

The model is built on the basis of algebraic operations of services. The model shows that the disadvantage of a web-service is that requests accumulate in a queue over time. This means that there will come a time when the queue will be full and the web-service will start to fail. To solve this problem, it is necessary to consider all possible ways of increasing the throughput of the web system.

To evaluate the effectiveness of the functioning of the pharmaceutical web-service developed using the proposed approach, the main indicators of its work were presented in comparison with the known pharmaceutical web-services. The analysis was carried out on the basis of information from independent resources: *www.sitechecker.pro*, *www.similarweb.com*, as well as the *Google Analytics* service. The following indicators were

considered and analyzed: time indicators spent on visiting a page, the number of pages viewed, the average bounce rate (resource: *www.similarweb.com*, *Google Analytics*), presented in Table 2. As well as performance indicators of the service (resource: *www.similarweb.com*) presented in Table 3. To analyze the parameters of the pharmaceutical web-service, the *Proekt* website was created on temporary domains.

According to the data presented, it can be concluded that the use of *Proekt* web-services for information aggregation has improved the quality indicators of services, and also from the analysis of the dynamics of changes in the service failure rate, we can conclude that the use of the *Proekt* web-service has improved the average service failure rate by 11 %.

Table 2 - Information on visit time, number and average bounce rate

Name of pharmaceutical services	Visit duration	Number of pages viewed	Bounce rate
Ventaltd.com.ua	3:44	3.54	52.60 %
Apteka24.ua	1:51	1.98	58.90 %
Aptekanizkihcen.ua	4:16	4.61	42.00 %
Apteka.net.ua/pharmacies	5:59	10.3	28.71 %
Medbrowse.com.ua/tabletki-cena/nikolaev	2:10	2.02	64.46 %
<i>Proekt</i>	5:31	8.64	32.41 %

Source: compiled by the authors

Table 3 - Information on performance indicators of web resources

Name of pharmaceutical services	First display of all content, s	Speed index, s	First simple CP, s	Maximum potential delay of the first administration, ms
ventaltd.com.ua	2.8	7.4	7.7	620
apteka24.ua	4.1	18.4	5.5	900
aptekanizkihcen.ua	5.4	9.2	8	890
apteka.net.ua/pharmacies	2	8.9	5.1	1610
medbrowse.com.ua/tabletki-cena/nikolaev	5.5	8.9	15.4	1970
<i>Proekt</i>	1.7	8.1	5.3	1648

Source: compiled by the authors

From the analysis of the dynamics of changes in the temporal performance indicators of web-services, we can conclude that the use of the proposed approach has improved the service speed index by 4 %.

CONCLUSIONS

The presented method of building and modelling composite web-services allows you to develop complex composite web-services. The method is based on an algebraic description of a web service based on a specialized algebra of services. The method consists of the following steps: description of the task of developing a web-service; development of the structure of a web-service based on the algebra of clarification of service operations; building models of functioning and interaction of service components; building simulation models of the composite service; defining the final structure of the composite service; web service implementation. The main tools for services; formal description of

the structure and creating composite services are service algebra and coloured Petri nets, implemented using CPN Tools. The process of building a composite model of a web-service based on coloured Petri nets is considered in detail. As an example of the application of the method, an example of building a pharmaceutical service is presented, in which the modelling of the interaction of web-services based on Petri nets is used. The main features of modelling complex composite web-services using coloured Petri nets are analysed. An analysis of the effectiveness of the functioning of a pharmaceutical web-service developed using the proposed approach was carried out. The main indicators of its work were presented in comparison with well-known pharmaceutical web-services. The time indicators of the service functioning and the performance indicators of the composite web service were analyzed. The analysis showed that the proposed approach to creating and modeling composite web-services is effective.

REFERENCES

1. Daneshpour, N., Maleki, K. N., Valipour, M. H. & Amirzafari, B. "A brief survey of software architecture concepts and service oriented architecture". *Proceedings - 2009 2nd IEEE International Conference on Computer Science and Information Technology*, 2009. p. 34–38. DOI: <http://dx.doi.org/10.1109/ICCSIT.2009.5235004>.
2. Yin, R. "Study of composing web-service based on SOA". *In Proceedings of the 2nd International Conference on Green Communications and Networks (GCN 2012)*. 2013; 2: 209–214. DOI: http://dx.doi.org/10.1007/978-3-642-35567-7_26.
3. Tiwari, A. & Mishra, V. K. "Colored Petri net-based techniques for constructing reliable web-service composition". *International Journal of Advance Research and Innovation*. 2018; 6 (1): 6–8.
4. Portchelvi, V., Venkatesan, V. P. & Shanmugasundaram, G. "Achieving web-services composition – a survey". *Softw. Eng.* 2012; 2 (5): 195–202. DOI: <http://dx.doi.org/10.5923/j.se.20120205.03>.
5. Zeshan, F. & Mohamad, R. "Semantic web-service composition approaches: overview and limitations". *International Journal of New Computer Architectures and their Applications*. 2011; 1 (3): 640–651.

6. Gómez, A., Rodríguez, R. J., Cambronero, M.-E. & Valero, V. “Profiling the publish/subscribe paradigm for automated analysis using colored Petri nets”. *Software & Systems Modeling. Springer*. 2019; 18 (5): 2973–3003. DOI: <https://doi.org/10.1007/s10270-019-00716-1>.
7. Klusch, M. “Semantic web-service description”. In book: *CASCOM: Intelligent Service Coordination in the Semantic Web. Birkhauser Basel publishing*. 2008. p. 31–57. DOI: http://dx.doi.org/10.1007/978-3-7643-8575-0_3.
8. Baryannis, G., Danylevych, O., Karastoyanova, D., Kritikos, K., Leitner, P., Rosenberg, F. & Wetzstein, B. “Service composition”. *Service Research Challenges and Solutions for the Future Internet. Part of the Lecture Notes in Computer Science book series (LNCCN, volume 6500).Springer*. 2010. p. 55–84.
9. Alcaraz-Mejia, M., Campos-Rodriguez, R. & Caballero-Gutierrez, M. “Modeling and simulation of task distribution using colored Petri nets”. 2017. DOI: <http://dx.doi.org/10.5772/67950>.
10. Freddo, F., Sawicki, S., Frantz, R.Z. & Roos-Frantz, F. “Using timed and colored Petri nets for modeling, simulation, and analysis of integration solutions”. *Preprint Int. Journal of Web Engineering and Technology*. 2019; 14(3): 231. DOI: <http://dx.doi.org/10.1504/IJWET.2019.105587>.
11. Jensen, K., Kristensen, L.M., Wells, L. “Colored Petri Nets and CPN tools for modelling and validation of concurrent systems”. *International Journal on Software Tools for Technology Transfer. Computer Science. Springer*. 2007; 9: 213–254. DOI: <http://dx.doi.org/10.1007/s10009-007-0038-x>.
12. Naji, H. A. H., Qingji Xue, Q. & Zheng, L. Zh. K. “A colored Petri nets based model and verification for services composition”. *ICEMCE 219. IOP Publishing. IOP Conf. Series: Journal of Physics: Conf. Series* 1314. 2019; 012144: 1-12. DOI: <http://dx.doi.org/10.1088/1742-6596/1314/1/012144>.
13. Deng, X., Lin, Z., Cheng, W., Xiao, B., Fang, L. & Li, L. “Modeling web-service choreography and orchestration with colored Petri nets”. *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. IEEE Computer society*. 2007. p. 838–843. DOI: <http://dx.doi.org/10.1109/SNPD.2007.486>.
14. Nie, K., Wang, H., Jing, X. & Xie, Z. “Web service automatic compilation model based on colored Petri net”. *Communications and Networking/* 2013; 5: 101–105. DOI: <http://dx.doi.org/10.4236/cn.2013.51B023>.
15. Kumar, P. & Gupta, R. “Dependency modeling of a SOA based system through colored Petri nets”. *CIT. Journal of Computing and Information Technology*. 2016; 24 (3): 253–269. DOI: <http://dx.doi.org/10.20532/cit.2016.1002787>.
16. Kallab, L., Mrissa, M., Chbeir, R. & Bourreau, P. “Using colored Petri nets for verifying RESTful service composition “. *Conference: 25th International Conference on Cooperative Information Systems*. At: Rhodes, Greece. Project: BEMServer – Building Monitoring and Data analysis for Energy Management. 2017. DOI: http://dx.doi.org/10.1007/978-3-319-69462-7_32.
17. Rostami, N. H., Kheirkhah, E. & Jalali, M. “Web services composition methods and techniques: A review”. *International Journal of Computer Science, Engineering and Information Technology (IJCEIT)*. 2013; 3 (6): 15–29. DOI: <http://dx.doi.org/10.5121/IJCSEIT.2013.3603>.
18. Niknama, M. & Karshenas, S. “Sustainable design of buildings using semantic BIM and semantic web-services.” *International Conference on Sustainable Design, Engineering and Construction*. Procedia Engineering 118. 2015. p. 909–917. DOI: <http://dx.doi.org/10.1016/j.proeng.2015.08.530>.
19. Sangers, J., Frasincar, F., Hogenboom, F. & Chepegin, V. “Semantic web-service discovery using natural language processing techniques”. *Expert Syst. Appl.* 2013; 31: 1–27.
20. Moghaddam, M. & Davis, J. G. “Service selection in web-service composition: A comparative review of existing approaches”. In *Web Services Foundations, First, A. Bouguettaya, Q. Z. Sheng and F. Daniel, Eds*. New York: Springer. 2014. p. 321–346. DOI: http://dx.doi.org/10.1007/978-1-4614-7518-7_13.
21. Ai, L. “QoS-Aware web-service composition using genetic algorithms”. *Queensland University of Technology*. 2011. 274 p.

22. Latorre, J. I. & Jiménez, E. “Simulation-based optimization of discrete event systems with alternative structural configurations using distributed computation and the Petri net paradigm”. *Simulation*, November 2013; 89 (11): 1310–1334. DOI: <https://doi.org/10.1177/0037549713505761>.
23. Valero, V., Masia, H., Diaz, G. & M. Cambronero, M.E. “Color Petri net modeling of the publish/subscribe paradigm in the context of web-service resources”. *International Seminar on Formal Methods for Industrial Critical Systems*. 2015; 9128: 81–95. DOI: https://doi.org/10.1007/978-3-319-19458-5_6.
24. Dasgupta, S. “A semantic framework for event-driven service composition”. A Dissertation in Computer Science and Telecommunications and Computer Networking Presented to the Faculty of the University of Missouri Kansas City”. University of Missouri Kansas City, 2011. *International Journal on Web Service Computing (IJWSC)*. 2016; Vol.7, No.1.
25. Kritikos, K., Kubicki, S. & Dubois, E. “Goal-based business service composition”. *Serv. Oriented Comput. Appl.* 2013; 7 (4): 231–257. DOI: <https://doi.org/10.1007/s11761-012-0126-y>.
26. Dworzański, L. W. & Lomazova, I. A. “CPN instrumental modeling and verification of nested Petri nets”. *Automatic control and computer science*. 2013; 47 (7): 393–402.
27. Hamadi R. & Benatallah B. “A Petri net based-model for web service composition”. *In proc. the 14th Australasian database conference, Adelaide. Darlinghurst: Australian Computer Society*. 2003. p. 191–200.
28. Mateo, J.A., Valero, V., Macià, H. & Díaz, G. “A coloured Petri net approach to model and analyses Stateful Workflows Based on WS-BPEL and WSRF”. B: Canal, C., Idani, A. (eds.) *SEFM 2014 Workshops*. 2015; 8938: 389–404. DOI: https://doi.org/10.1007/978-3-319-15201-1_26.
29. Jensen K. “Colored Petri nets: basic concepts, methods of analysis and practical use”. *Part of the book series: Monographs in Theoretical Computer Science. An EATCS Series (EATCS), Springer Science and Business Media*. 2013.
30. Westergaard, M. “CPN Tools 4: Multiformalism and extensibility. In the application and theory of Petri nets and parallelism”. *Springer*. Berlin: Heidelberg. 2013. p. 400–409. DOI: https://doi.org/10.1007/978-3-642-38697-8_22.
31. Peng, Z. X. & Jun, Z. “The web-service composition based on colored Petri net”. *International Journal of Research in Engineering and Science (IJRES)*. ISSN (Online): 232-9364. 2017; 5 (3): 39–44.
32. Gozhyj A., Kalinina I., Gozhyj V. & Danilov V. “Approach for modeling search web-services based on color Petri nets”. *In: Data Stream Mining & Processing. DSMP 2020. Communications in Computer and Information Science. Springer, Cham*. 2020; 1158: 525–538. DOI: https://doi.org/10.1007/978-3-030-61656-4_35.
33. Gozhyj A., Kalinina, I., Gozhyj, V. & Vysotska V. “Web-service interaction modeling with colored Petri nets”. *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Metz. 2019. p. 319–323. DOI: <https://doi.org/10.1109/IDAACS.2019.8924400>.

Conflicts of Interest: the authors declare no conflict of interest

Received 22.08.2022

Received after revision 12.10.2022

Accepted 19.10.2022

DOI: <https://doi.org/10.15276/aait.05.2022.14>

УДК 004.75.02

Метод розробки та моделювання композитних веб-сервісів

Гожий Олександр Петрович¹⁾

ORCID: <https://orcid.org/0000-0002-3517-580X>; alex.gozhyj@gmail.com. Scopus Author ID: 57198358626

Калініна Ірина Олександрівна¹⁾

ORCID: <https://orcid.org/0000-0001-8359-2045>; irina.kalinina1612@gmail.com. Scopus Author ID: 57198354193

Гожий Віктор Олександрович²⁾ORCID: <https://orcid.org/0000-0002-5341-0973> ; gozhyi.v@gmail.com. Scopus Author ID: 57198357517¹⁾ Чорноморський національний університет імені Петра Могили, вул. 68 Десантників, 10. Миколаїв, 54000, Україна²⁾ IT company “Template Monster”, вул. Чкалова 99А. Миколаїв, 54056, Україна

АНОТАЦІЯ

Стаття присвячена методу розробки та моделювання композитних веб-сервісів. Композиція веб-сервісів використовується для отримання нових функціональних можливостей на основі взаємодії існуючих веб-сервісів. Побудова композитних веб-сервісів відбувається у кілька етапів: специфікації (визначення типу сервісу); розробка структури сервісу на основі алгебри сервісів; моделювання композиції сервісів; вибору варіанта сервісу та генерації сервісу. Головними елементами пропонованого підходу є алгебра сервісів та моделі взаємодії веб-сервісів (базові та композитні). У наведеному підході формально представлений облік основних аспектів при вирішенні проблем, пов'язаних із побудовою ефективних композитних веб-сервісів та підбором математичних моделей, а саме: опис веб-сервісів, визначення структури веб-сервісів, облік динаміки зміни інформації, облік основних невизначеностей при побудові структури web-сервісів. На підставі підходу розроблено метод побудови композитних веб-сервісів. В основу методу покладено опис алгебри веб-сервісів на основі спеціалізованої алгебри сервісів. Метод складається з наступних етапів: опис завдання розробки веб-сервісу; розробка структури веб-сервісу на основі алгебри сервісів; формальний опис структури та уточнення операцій сервісу; побудова моделей функціонування та взаємодії компонентів сервісу; побудова імітаційних моделей композитного сервісу; визначення остаточної структури композитного сервісу; реалізація веб-сервісу. Основним інструментом опису композитних сервісів є алгебра сервісів, а інструментом побудови моделі – кольорові мережі Петрі, реалізовані з допомогою CPN Tools. Як приклад застосування розробленого методу описано побудову фармацевтичного сервісу, в якому використано моделювання взаємодії веб-сервісів на основі мереж Петрі. Розглянуто процес побудови композитної моделі веб-сервісу. Процес починається із побудови структури моделі фармацевтичного веб-сервісу на основі алгебри сервісів. З алгебри сервісів розробили оператори для реалізації композитних сервісів. Розроблено структуру моделі, яка подається у вигляді формального опису. Детально описаний процес побудови моделей фармацевтичного сервісу серед CPN Tools.

Ключові слова: композитний веб-сервіс; алгебра сервісів; структура сервісу; модель композитного сервісу; кольорові мережі Петрі

ABOUT THE AUTHORS



Aleksandr P. Gozhyj – Doctor of Engineering Sciences, Professor, Professor of Intelligent information systems Department. Petro Mohyla Black Sea National University, 10, 68 Desantnykiv Str. Mykolaiv, 54000, Ukraine

ORCID: <https://orcid.org/0000-0002-3517-580X>; alex.gozhyj@gmail.com. Scopus Author ID: 57198358626

Research field: Artificial intelligence methods; machine learning; probabilistic-statistical methods; intelligent web technology

Гожий Олександр Петрович – доктор технічних наук, професор, професор кафедри Інтелектуальних інформаційних систем Чорноморський національний університет імені Петра Могили, вул. 68 Десантників, 10, Миколаїв, 54000, Україна



Irina A. Kalinina – PhD, Associate Professor, Associate Professor of Intelligent information systems Department. Petro Mohyla Black Sea National University, 10, 68 Desantnykiv Str. Mykolaiv, 54000, Ukraine

ORCID: <https://orcid.org/0000-0001-8359-2045>; irina.kalinina1612@gmail.com. Scopus Author ID: 57198354193

Research field: Probabilistic-statistical methods; intelligent web technology; machine learning; coloured Petri nets

Калініна Ірина Олександрівна – кандидат технічних наук, доцент, доцент кафедри Інтелектуальних інформаційних систем Чорноморський національний університет імені Петра Могили, вул. 68 Десантників, 10, Миколаїв, 54000, Україна



Victor A. Gozhyj – PhD, Chief operating officer (COO) TemplateMonster, 99A, Chkalov Str. Mykolaiv, 54056, Ukraine

ORCID: <https://orcid.org/0000-0002-5341-0973> ; gozhyi.v@gmail.com. Scopus Author ID: 57198357517

Research field: Modern web technology; coloured Petri nets

Гожий Віктор Олександрович – кандидат технічних наук, головний операційний директор (COO) Template Monster, вул. Чкалова, 99А. Миколаїв, 54056, Україна