

УДК 681.513

Мочалов Валерий Петрович, Яковлев Сергей Владимирович,  
Братченко Наталья Юрьевна

## РАЗРАБОТКА МЕТОДИКИ ПОСТРОЕНИЯ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ТЕЛЕКОММУНИКАЦИОННЫМИ СЕТЯМИ НА ОСНОВЕ ТЕХНОЛОГИИ CORBA<sup>1</sup>

*В статье приводится методика построения распределенной системы управления сетями связи на основе технологии CORBA. Описаны функциональные компоненты среды CORBA. С использованием формализма цветных сетей Петри разработана модель интеграции программных компонент.*

**Ключевые слова:** телекоммуникации, CORBA, программный компонент, процессорный модуль, объектный брокер, интеграция.

**Mochalov Valery P., Yakovlev Sergey V., Bratchenko Natalia Yu.**  
**METHODOLOGY FOR BUILDING DISTRIBUTED CONTROL SYSTEM FOR TELECOMMUNICATION NETWORKS BASED ON CORBA TECHNOLOGY**

*The article offers a description of methodology for building a distributed system for managing communication networks based on the CORBA technology. There is a view on the functional components of the CORBA environment. The formalism of Petri colour nets was used here to develop a model for integrating programme components.*

**Key words:** telecommunication, CORBA, programme component, processing module, object broker, integration

Реализованные системы управления телекоммуникационными сетями и услугами (TMN, OSS/BSS и др.) имеют существенные ограничения, сложно интегрируются с фрагментами сетей и не в полной мере соответствуют требованиям, предъявляемым к управлению качеством [1]. Устранить данные недостатки возможно путем построения распределенных систем управления (PCY).

Функциональная модель распределенной системы управления представлена на рис. 1 в виде сетевой модели, состоящей из  $N$  узлов процессорных модулей (ПМ),  $m$  транзитных узлов (ТУ),  $k$  узлов диспетчеров. В состав внешнего окружения PCY входят сетевые узлы различного типа и назначения, со стороны которых поступает множество входных сигналов  $X = \{X_1, \dots, X_N\}$ , в качестве реакции PCY вырабатывает множество выходных сигналов, выдаваемых во внешнее окружение. Элементами коммутационного оборудования являются коммутационные блоки, число и разновидность которых для конкретных PCY могут быть различными. Задание, выполняемое PCY, представляется в виде множества  $U = \{U_i\}$ ,  $i = \overline{1, K}$ , программных компонентов (ПК), которые связаны между собой как зависимостью по данным, так и логикой выполнения. Распределение ПК по  $N$  процессорным модулям PCY описывается двоичной матрицей  $X = \|X_{ij}\|$ ,  $i = \overline{1, K}$ ,  $j = \overline{1, N}$ , где  $X_{ij} = 1$ , если ПК  $P_i$  размещается в ПМ  $N_j$ ,  $X_{ij} = 0$  – в противном случае. Каждый ПК  $U_i$ ,  $i = \overline{1, L}$ , размещен только в одном ПМ, т. е.

$$\sum_{j=1}^N X_{ij} = 1,$$

$i = \overline{1, L}$ , а в каждом ПМ  $M_j$ ,  $j = \overline{1, N}$  размещен по крайней мере один ПК и не может быть размещено более чем  $L$  ПК, т. е.

$$1 \leq \sum_{i=1}^L X_{ij} = L, \quad j = \overline{1, N}.$$

<sup>1</sup> Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 13-07-00130 «Разработка концепции процессно-ориентированного управления телекоммуникационными сетями и услугами».

Для любого задания  $U = \{U_i\}$  может быть найдено минимальное время  $\tau_m$  его выполнения. Достижение значения  $\tau_m$  гарантируется в такой системе, где каждому ПМ назначен для решения только один ПК и, следовательно, число ПМ равно числу ПК, т. е.  $N = L$ . Величина времени выполнения задания  $\tau$  оценивается для каждого заданного числа ПМ и каждого заданного распределения ПК с учетом времени ожидания каждым из ПК необходимых ему входных данных. При этом рассматривается только процесс однократного выполнения задания, т. е. такой процесс вычислений, при котором однократно выполняется каждая из процедур ПК и который начинается с выполнения некоторого начального ПК и заканчивается выполнением некоторого множества конечных ПК.

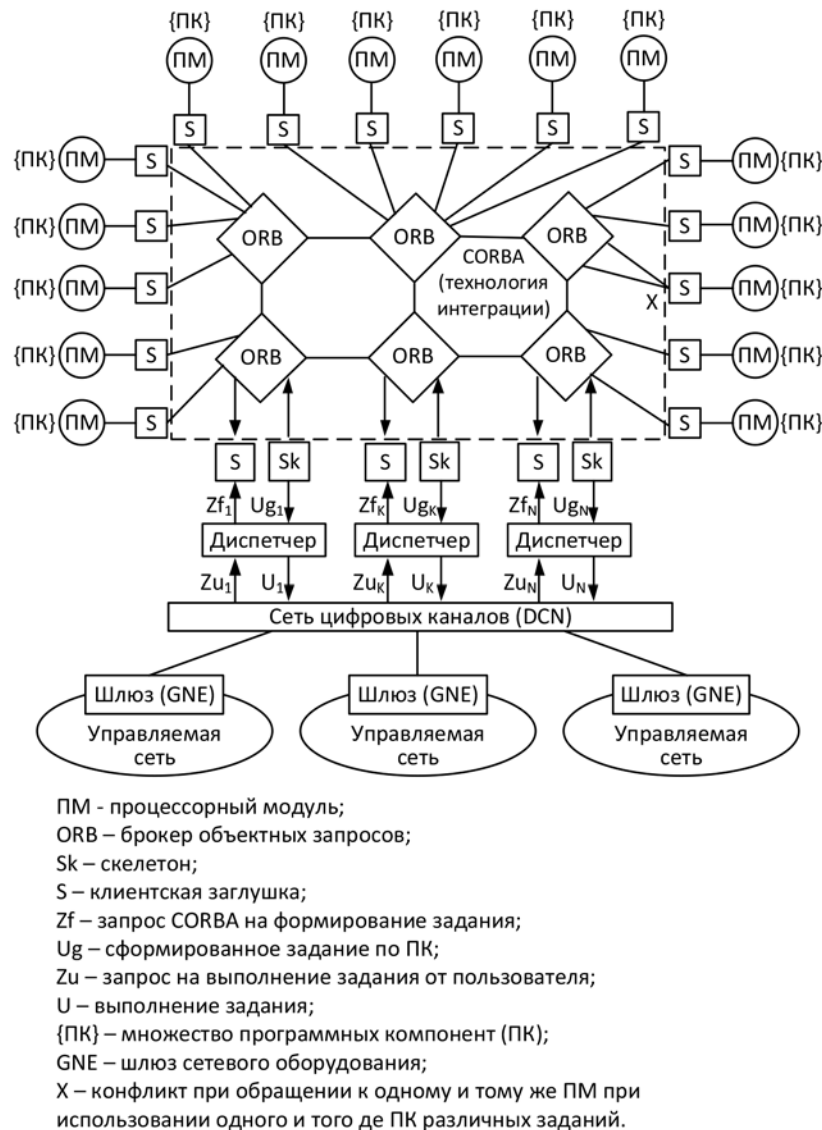


Рис. 1. Модель распределенной системы управления на основе CORBA

Порядок выполнения ПК определяется на множестве  $\Omega$  функцией  $E(U_i) = \omega_i$ , задающей для каждого ПК  $U_i$  подмножество  $\omega_i \in \Omega$  таких ПК, которые генерируют входные данные для этого ПК и, следовательно, должны быть выполнены до начала его выполнения. Предполагается также, что

входные данные ПК  $U_i$ , являющиеся выходными данными ПК  $U_k \in \omega_i$ , могут быть переданы от каждого из ПК  $U_k$  к ПК  $U_i$  только после завершения выполнения соответствующего ПК  $U_k$  [2]. Для каждого ПК  $U_i \in \Omega$  известно время его выполнения  $\tau_i$  и время передачи данных между ПК.

В качестве показателя производительности РСУ примем время  $\tau$  выполнения задания, т. е. время от момента начала выполнения начального ПК множества  $\Pi$  до момента завершения выполнения конечных ПК. Величина  $\tau$  оценивается для каждого заданного числа ПМ и каждого распределения ПК с учетом времени ожидания каждым из ПК  $U_i \in \Omega$  необходимых ему входных данных. Качество функционирования РСУ оценивается набором значений времени выполнения всех процессов задания, т. е.  $T = \{\tau_i\}$ ,  $i = \overline{1, L}$ . Требуется найти минимальное число  $N$  ПМ, распределение ПК  $X$  и расписание  $R$  их выполнения при целевой допустимости времени выполнения задания  $\tau(N, S_0, X, R) \leq \tau_{\text{дон}}$ .

Неопределенность состояния элементов системы существенно затрудняет выполнение правильного выбора варианта ее построения, оценку этих показателей, приводит к необходимости разработки инженерной методики и инструментальных средств, ориентированных на анализ ее вероятностно-временных характеристик (ВВХ).

На каждом сетевом элементе имеется набор ПК, реализующий элементы алгоритмов приложений. Наборы ПК на каждом из узлов сети могут отличаться. РСУ включает набор диспетчеров, выполняющих планирование процесса реализации управляющих воздействий, принимающих запросы на реализацию услуг связи, формирующих управляющие воздействия на элементы сети. Круг решаемых системой задач ограничен только набором ПК [3].

Процесс функционирования РСУ определяется процедурами взаимодействия между элементами системы: распределения заданий по ПК; интеграции ПК; маршрутизации; алгоритмов управления сетью сигнализации и сетевыми элементами.

На первом этапе диспетчеру поступает запрос  $ZU$  на реализацию задания  $U$ . Диспетчер формирует запрос  $Zf$  на формирование задания и определения распределенных объектов (ПК). Связь между диспетчером и ПМ устанавливается через функциональные компоненты среды CORBA, представленные на рис. 2.

На втором этапе от диспетчера поступает запрос в каждый ПМ, потенциально реализующих задание. После получения ответов от ПМ выполняется запрос об устранении конфликтов при использовании одного и того же ПК различными видами заданий. При возникновении конфликтной ситуации выполняется процедура разрешения конфликтов.

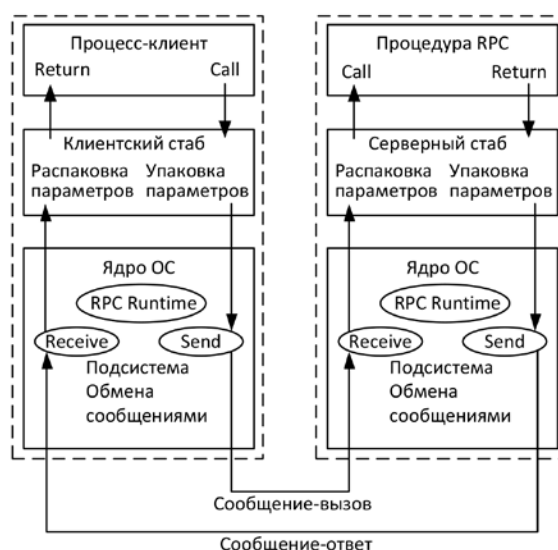


Рис. 2. Функциональные компоненты среды CORBA

На третьем этапе в среде CORBA выполняется процедура объединения ПК в соответствии с зависимостью по данным и логикой выполнения.

Предлагаемый алгоритм функционирования представлен на рис. 3. Задача распределения ПК по ПМ решается в соответствии с алгоритмом, представленным на рис. 4.

Считаем, что имеется  $n$  ПК  $f_1, \dots, f_n$  и  $d$  ПМ. Для любой пары ПК  $f_i$  и  $f_j$  известна частота  $P(i, j)$  с которой к ним происходят соседние запросы.

Размещение  $n$  ПК по  $d$  ПМ задается разбиением множества всех ПК на  $d$  групп  $\Phi = \{F_1, F_2, \dots, F_d\}$ , так, что

$$\bigcup_{i=1}^n f_i = \bigcup_{i=1}^d F_i \text{ и } F_k \cap F_l = \emptyset, k \neq l. \quad (1)$$

Средняя частота возникновения конфликтов  $C_k$  на одном ПМ с номером  $k$  равна:

$$C_k = \sum_{i,j} p(i, j). \quad (2)$$

Суммарная частота возникновения конфликтов  $C$  при разбиении  $\Phi$  равна:

$$C = \sum_{k=1}^d C_k = \sum_{k=1}^d \sum_{i,j} p(i, j). \quad (3)$$



Рис. 3. Алгоритм функционирования РСУ

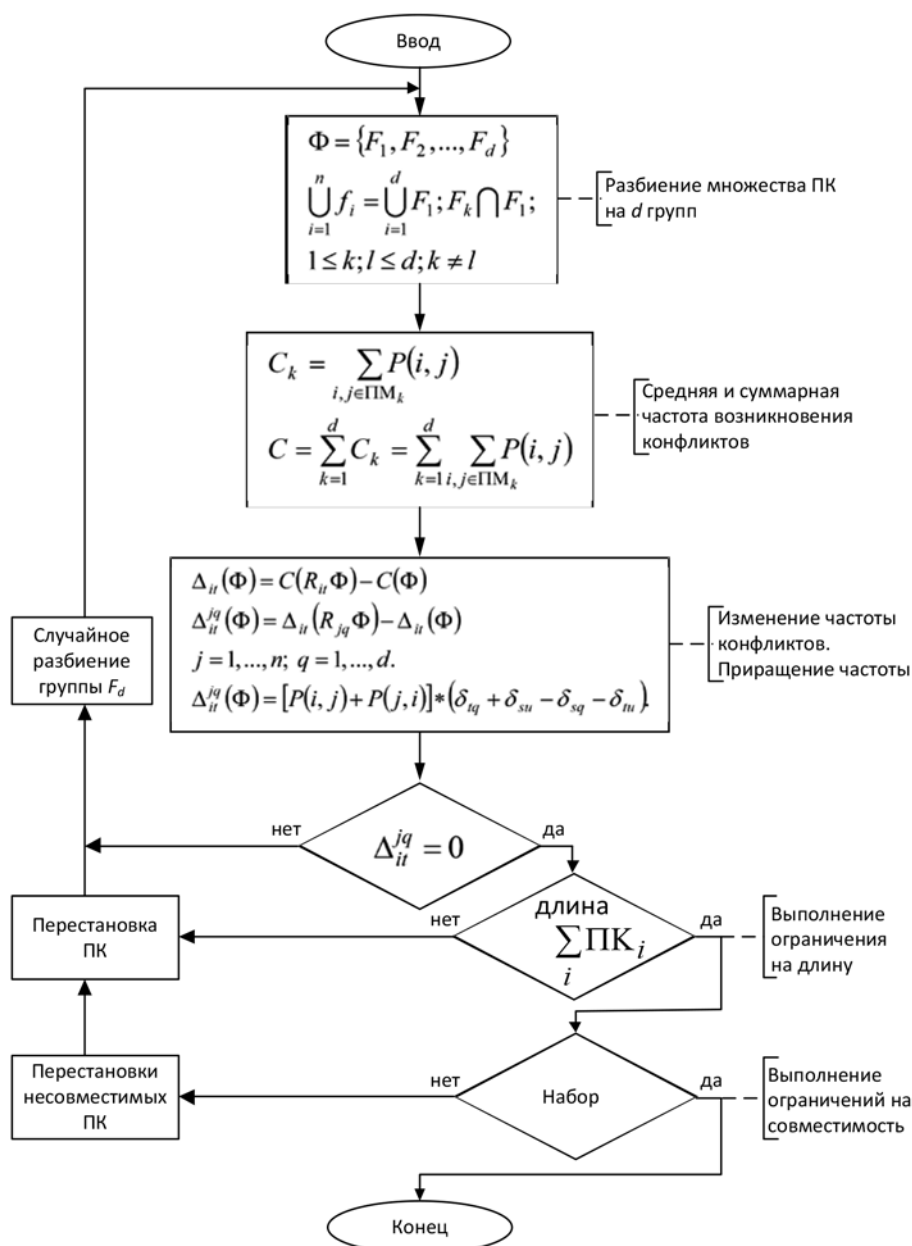


Рис. 4. Схема алгоритма субоптимального распределения ПК по ПМ

Оптимальное разбиение  $\Phi$  минимизирует критерии  $C$ . Алгоритм локальной оптимизации критерия состоит из последовательности одинаково построенных шагов. Изменение значения критерия при переносе  $PK_i$  из группы  $F_s$  в группу  $F_l$  подсчитывается по формуле

$$\Delta C = \sum_{i \in \Pi M_l} (P(i, j) + P(j, i)) - \sum_{i \in \Pi M_s} (P(i, j) + P(j, i)) \quad (4)$$

Этот алгоритм за конечное число шагов сходится к точке локального минимума, при этом любой перенос файлов из группы в группу не приводит к увеличению критерия. При большом числе ПК вычисление величин  $\Delta C$  на каждом шаге оказывается трудоемким. В соответствии с методом, изложенным в приращения  $\Delta C$  могут пересчитываться более эффективным рекуррентным способом.

При этом для упорядочения всевозможных способов разбиения элементов  $ПК_i, i=1, \dots, n$  на  $d$  групп вводят систему операторов  $R = \{R_{it}, i=1, \dots, n; t=1, \dots, d\}$ , означающих перенос  $ПК_i$  в класс  $t$ , т.е. выполняют операцию  $R_{it}\Phi$ . Тогда  $\Delta_{it}(\Phi) = C(R_{it}\Phi) - C(\Phi)$ , где  $\Delta_{it}(\Phi)$  – приращение критерия (3).

Данное выражение позволяют реализовать следующий рекуррентный способ вычисления  $\Delta_{it}^{iq}$ . Обозначим через  $\Delta_{it}^{iq}(\Phi)$  приращение величин  $\Delta_{it}(\Phi)$  при переходе к новому разбиению  $R_{jq}(\Phi)$ , где  $R_{jq} \in R$ :

$$\Delta_{it}^{jq}(\Phi) = \Delta_{it}(R_{jq}\Phi) - \Delta_{it}(\Phi), \quad i=1, \dots, n; \quad q=1, \dots, d.$$

Тогда

$$\Delta_{it}(R_{jq}\Phi) = \Delta_{it}(\Phi) + \Delta_{it}^{jq}(\Phi);$$

$$\Delta_{it}^{jq}(\Phi) = [P(i, j) + P(j, i)](\delta_{iq} + \delta_{su} - \delta_{sq} - \delta_{iu}),$$

где  $s, u$  – номера групп, в которых находились  $ПК_i$  и  $ПК_j$  при разбиении;  $t, q$  – номера групп, в которые перенесены эти ПК;  $\delta_{iq}$  – символ Кронекера

$$\delta_{iq} = \begin{cases} 1, & \text{если } t=q; \\ 0, & \text{если } t \neq q. \end{cases}$$

Считаем, что последовательность запросов к ПК описывается процессом независимых испытаний с распределением

$$(P_1, P_2, \dots, P_n), \sum_i P_i = 1.$$

Для анализа процедуры интеграции ПК предлагается использовать формализмы раскрашенных сетей Петри (CPN). Соответствующая модель представлена на рис. 5.

Опишем параметры данной сети:

*Множество цветов*

$REQ = (\text{Number}, \text{Data})$ , где Number – целое неотрицательное число;

Data – строковый тип.

$KEY = (Key_1, Key_2, \dots, Key_x)$ , где  $Key_i (i \in [1, x])$  – целое неотрицательное число.

$REP = (\text{Data}, KEY)$ .

$SET = (\text{Number}, KEY)$ .

$PC = (Key_i, pc_i)$ , где  $pc_i$  – двоичный набор данных.

$SetPC = (pc_1, pc_2, \dots, pc_x)$ .

*Множество позиций:*

- $P_{vx}$  – позиция, наличие фишки в которой свидетельствует о наличии запроса на реализацию задания. Фишки в этой позиции характеризуются цветом REQ.
- $Repository$  – позиция, моделирующая хранилище ключей соответствующие ПК, в которой содержатся фишки цветом REP.
- $P_{кл}$  – позиция в которой содержатся фишки, определяющие ключи соответствующие ПК. Фишки в этой позиции характеризуются цветом SET.
- $ПМ_1, ПМ_2, \dots, ПМ_M$  – позиции, моделирующие ПМ и содержащие ПК. Фишки в этой позиции характеризуются цветом PC.
- $ORB_1, ORB_2, \dots, ORB_L$  – позиции фишки которых определяют набор ПК и характеризуются цветом PC.
- $P\phi_1, \dots, P\phi_L$  – позиции, хранящие фишки цветом SET.
- $P_{вых_1}, \dots, P_{вых_L}$  – позиции, фишки в которых определяют результат работы механизма интеграции ПК. Цвета фишек позиции определяются как SetPC.

- $t_l$  – переход выполняет анализ запроса как реализацию задания. Согласно значению Data из совокупности фишек, хранящихся в позиции Repository, выбирается фишка цветом REP, помещаемая в позицию Ркл.
- $\{t1l \dots t1L\}, \{t2l \dots t2L\}, \dots, \{tMl \dots tML\}$  – переходы, моделирующие процесс выборки ПК, хранящихся на ПМ. В качестве требования к срабатыванию перехода выступает наличие фишек с соответствующим значением Key во входных позициях ПМ и Ркл.
- $t\phi_P, \dots, t\phi_L$  – переходы срабатывают в случае наличия фишки цветом SET в позиции Ркл, причем компонента Number определяет номер  $t\phi$ .
- $tORB_P, \dots, tORB_L$  – переходы, моделирующие формирование задания из отдельных ПК. Требованием к срабатыванию перехода является наличие в соответствующих входных позициях ORB и Рф фишек с одинаковым значением Key. После срабатывания перехода в выходные позиции Рвых помещаются фишки цветом SetPC. Задержки на переходах сети указаны на рис. 5.

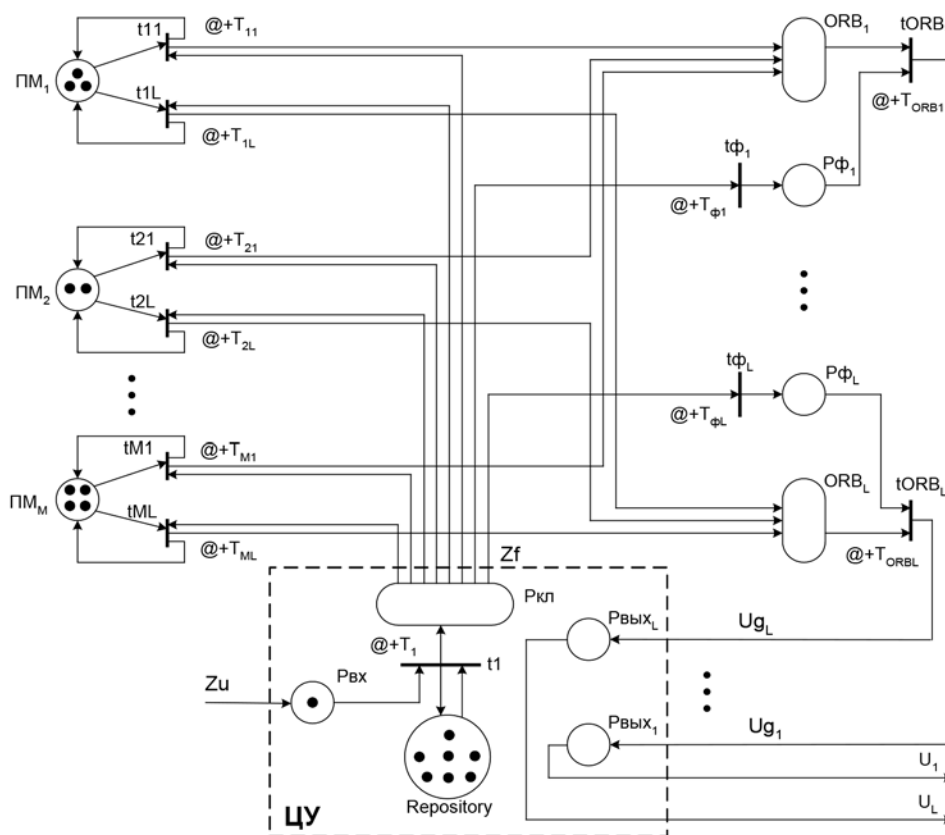


Рис. 5. Сеть Петри механизма интеграции ПК

Поступлении запроса  $Zu$  представляется помещением фишки цветом  $REQ = (Number, Date)$  в позицию  $P_{vx}$ . Первая компонента представляет собой порядковый номер поступления запроса. Вторая компонента содержит характеристику запроса, которая используется при выборе ресурсов из позиции Repository. В результате этого в позиции  $P_{kl}$  по истечении времени  $T_1$  появляется фишка с параметрами  $SET = (Number, KEY)$ , где компонента  $KEY$  представляет собой совокупность ключей  $Key_i$ , которые используются для выборки ПК. По истечении интервалов времени, необходимых для взаимодействия с ПМ, фишки цветом  $PC$ , соответствующие ПК, изымаются из позиции ПМ и поме-



шаются в позиции ORB, согласно, параметру Number. Одновременно фишки цветом SET поступают в соответствующие позиции Рф. Компонента KEY фишек в позиции Рф, определяет порядок интеграции отдельных ПК, которые моделируются фишками, находящимися в позициях ORB. Результат интеграции ПК в виде фишек SetPC появляется после задержки по времени TORB в позициях Рвых.

Учитывая различные гипотезы о законах распределения вероятностей событий и время их выполнения, возможно получение вероятностно-временных характеристик моделируемого процесса.

#### *Литература*

1. Мочалов В. П. Формализованное представление процесса анализа системы обслуживания заказов оператора связи / В. П. Мочалов, Г. В. Слюсарев, С. В. Яковлев, С. В. Жубреев, А. А. Кульпинов, Д. В. Мочалов // Наукоемкие технологии. 2012. № 7. С. 66–70.
2. Mochalov V., Bratchenko N. Call management model in telecommunications control system // Scientific enquiry in the contemporary world: theoretical basics and innovative approach, B&M Publishing. San Francisco, California. 2014. С. 143–151.
3. Mochalov V., Bratchenko N. The distributed system of gathering and the control of parameters of object // Pressing issues and priorities in development of the scientific and technological complex. CA, USA, B&M Publishing, 2013. С. 137–146.

УДК: 681.3

**Орлова Анна Юрьевна, Сорокин Александр Анатольевич**

## **ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ ДЕЯТЕЛЬНОСТИ СИСТЕМЫ E-LEARNING ПОСРЕДСТВОМ ПЛАТФОРМ IBM WEBSHERE BUSINESS MODELER И ANYLOGIC**

*В статье рассмотрены проблемы моделирования деятельности системы e-learning с применением платформ IBM WebSphere Business Modeler и AnyLogic и произведена оптимизация деятельности системы, что привело к разгрузке сервера, и как следствие к уменьшению потерь заявок.*

**Ключевые слова:** e-learning, BPMN-модель, дискретно-событийное моделирование, поддержка принятия решений, имитационное моделирование, IBM WebSphere Business Modeler, AnyLogic.

**Orlova Anna Yu., Sorokin Alexander A.**

### **OPTIMIZATION AND MODELING OF E-LEARNING SYSTEM THROUGH PLATFORMS IBM WEBSHERE BUSINESS MODELER AND ANYLOGIC**

*The article offers a view on the issues related to modeling the performance of the system e-learning employing the platforms IBM WebSphere Business Modeler and AnyLogic, as well as presents optimization of the system's performance, which resulted in server relief leading to a decrease in the request loss.*

**Key words:** e-learning, BPMN-model, discrete event-based modeling, decision-making support, simulation, IBM WebSphere Business Modeler, AnyLogic.

E-learning в современном образовательном процессе играет с каждым годом все более значимую роль. Подобная инфокоммуникационная система, основанная на архитектуре клиент-сервер, позволяет посредством web-браузера предоставлять пользователям учебные курсы с возможностью создания собственного пользовательского интерфейса, что упрощает процесс восприятия информации. Многие организации из системы высшего образования внедрили или начинают внедрять подобные e-learning системы, однако остается открытым вопрос об эффективности их функционирования.