

## Электронное моделирование

УДК 004.021+656.25

**П. Е. Булавский, д-р техн. наук,  
Д. С. Марков, канд. техн. наук,  
В. Б. Соколов, канд. техн. наук,  
Т. Ю. Константинова, канд. техн. наук**

Кафедра «Автоматика и телемеханика на железных дорогах»,  
Петербургский государственный университет путей сообщения  
Императора Александра I

### **ФОРМАЛИЗАЦИЯ АЛГОРИТМИЧЕСКОГО ОПИСАНИЯ СИСТЕМ ОБЕСПЕЧЕНИЯ ЖИЗНЕННОГО ЦИКЛА ЖЕЛЕЗНОДОРОЖНОЙ АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ**

В статье рассмотрены задачи синтеза алгоритмического отображения систем обеспечения жизненного цикла железнодорожной автоматики и телемеханики как сложных систем массового обслуживания. Принято, что алгоритмическое отображение представляется в виде алгоритмов обслуживания заявок, должно формализоваться на языке параллельных логических схем алгоритмов (ПЛСА) и использоваться в качестве основы имитационных моделей сложных систем массового обслуживания. Показана целесообразность синтеза объединенного алгоритма обслуживания заявок. Дано определение объединенного алгоритма обслуживания заявок. Разработан метод объединения алгоритмов, основанный на построении матрицы вхождений элементов исходного алфавита ПЛСА, анализе разбиений по ее составляющим, позволяющий синтезировать объединенный алгоритм обслуживания заявок при однократном вхождении элементов исходного алфавита, минимальном количестве разделительных логических условий и минимизированном количестве разделительных логических условий. Применение предложенного метода объединения алгоритмов позволяет сократить объем имитационных моделей и соответственно количества ошибок в моделирующей программе.

системы обеспечения жизненного цикла железнодорожной автоматики и телемеханики; сложные системы массового обслуживания; имитационное моделирование; алгоритм обслуживания заявок; объединенный алгоритм; матрица вхождений; разделительные логические условия

### **Введение**

Жизненный цикл железнодорожной автоматики и телемеханики (ЖАТ) – это совокупность взаимосвязанных, последовательно выполняемых процессов от формирования концепции безопасности и исходных требований к устройству или системе железнодорожной автоматики и телемеханики до вывода из экс-

платации и утилизации. Жизненный цикл ЖАТ включает следующие основные процессы:

- разработка концепции безопасности и соответствующих документов;
- разработка аппаратно-программных решений и конструкторско-технической документации;
- доказательство безопасности принятых технических, программных решений и сертификация на безопасность;
- тиражирование – проектирование аппаратных и программных средств ЖАТ для конкретных объектов железнодорожного транспорта;
- изготовление аппаратно-программных средств ЖАТ по проекту;
- выполнение пусконаладочных работ на объекте, проведение соответствующих испытаний и приемка в эксплуатацию;
- техническая эксплуатация и модернизация ЖАТ в процессе эксплуатации;
- выполнение процедур вывода ЖАТ из эксплуатации и утилизации.

Безопасность и бесперебойность движения поездов определяются не только потребительскими свойствами, показателями безотказности и безопасности ЖАТ как изделий, но и в определяющей степени качеством систем обеспечения жизненного цикла (СОЖЦ) ЖАТ, к которым относятся:

- комплекс методов и инструментальных средств доказательства безопасности ЖАТ [1–3];
- системы автоматизированного проектирования (САПР) ЖАТ [4, 5];
- системы диагностирования и удаленного мониторинга (СДУМ) технического состояния ЖАТ [6–9];
- системы электронного документооборота технической документации ЖАТ [10–12];
- автоматизированные системы управления хозяйством автоматики и телемеханики – АСУ-Ш [13].

Разработка и модернизация СОЖЦ ЖАТ представляют собой непрерывные процессы принятия и реализации сложных системотехнических решений, включающие методы и средства оценки их эффективности. В [14, 15] показано, что процессы, выполняемые СОЖЦ, относятся к дискретным в непрерывном времени, а СОЖЦ представляют собой сложные системы массового обслуживания (ССМО). В качестве методов и средств оценки эффективности применяются методы имитационного моделирования и имитационные модели (ИМ) [14, 16]. В качестве инструментального средства реализации ИМ СОЖЦ ЖАТ традиционно используется моделирующая среда GPSS World [17].

## 1 Основные положения

В [18] предложена обобщенная формализованная схема (ОФС) СОЖЦ ЖАТ как ССМО, включающая:

- модель внешней среды в виде вектора технологической нагрузки – потока заявок на обслуживание;
- структурно-алгоритмическое описание системы и процесса обслуживания заявок;
- параметрическое описание, включающее вероятностно-временные характеристики внешней среды и процесса обслуживания;
- множество операционных характеристик СОЖЦ ЖАТ.

В данной работе рассматривается алгоритмическое описание процессов обслуживания заявок, которое, в соответствии с [15], является основой синтеза имитационных моделей СОЖЦ ЖАТ. В качестве заявок, в зависимости от вида СОЖЦ ЖАТ, выступают: комплекты технических документов, информационные сообщения, задачи по экспертизе или проектированию систем и т. п. В [15] для алгоритмического отображения СОЖЦ ЖАТ предложено использовать язык параллельных логических схем алгоритмов (ПЛСА). При этом алфавит ПЛСА СОЖЦ ЖАТ включает следующие элементы: операторы  $o_i, i=1, I$  и логические условия: ждущие  $q_k, Q \in; k=1, K$ ; собственные  $g_s, s=1, S$ ; вероятностные  $\alpha_n, n=1, N$ ; операторы распараллеливания  $R \uparrow^p$ ; объединения  $\downarrow \&$  и тождественно-ложного логического условия  $w \uparrow^p$ . Правила записи ПЛСА с использованием предложенного алфавита изложены в [19].

Для алгоритмического отображения СОЖЦ ЖАТ как ССМО характерно большое количество вхождений операторов  $o_i, i=1, I$  и логических условий  $q_k, k=1, K$ ;  $g_s, s=1, S$ ;  $\alpha_n, n=1, N$  при сравнительно небольшом количестве их типов. В связи с этим целесообразно поставить задачу сокращения вхождения однотипных операторов и логических условий в алгоритмическое описание ССМО. Задача решается в целях сокращения объема, времени подготовки и отладки модели, количества ошибок в моделирующей программе, увеличения быстродействия GPSS-моделей. Для решения необходимо объединить алгоритмы обслуживания заявок  $A_j$  (АОЗ).

ССМО обслуживает множество заявок  $h_j \in H, j=1, J$  различного типа. Тип заявки  $h_j$  идентифицируется по выполняемому АОЗ. Каждая заявка обслуживается по своему алгоритму  $A_j, j=1, J$ , формализованная запись которого реализуется на ПЛСА с использованием предложенной выше символики:  $O, Q, G, A$ , т. е. описывает АОЗ в терминах операторов, соответствующих операциям, выполняемым ССМО.

В дальнейшем в тех случаях, когда не требуется различать типы членов алфавита, они обозначаются  $z_\beta, \beta=1, B$ , где  $B$  – общее количество элементов описания АОЗ. При объединении АОЗ необходимо учитывать, что после выполнения какого-либо  $z_\beta$  в различных алгоритмах  $A_j, j=1, J$  могут выполняться неодинаковые последовательности элементов. Для определения порядка выполнения элементов в объединенном алгоритме  $A_o$  при обслуживании заявок различного типа  $h_j, j=1$  вводятся дополнительные разделительные логические условия (РЛУ) –  $r$ . РЛУ ставятся в  $A_o$  после элемента  $z_\beta$ , образуя дерево проверок, число

выходов которого равно количеству неодинаковых элементов, выполняемых в различных  $A_j, j=1, J$  после  $z_\beta$ . Тип заявки  $j$  определяет элемент, который должен выполняться в алгоритме  $A_o$  после элемента  $z_\beta$ , и, следовательно, РЛУ представляет собой булеву функцию от типа заявок  $-f(j)$ . Для записи такой функции вводятся булевы переменные вида:

$$x_1 = \begin{cases} 1 - \text{если } j = 1; \\ 0 - \text{если } j \neq 1; \end{cases} x_2 = \begin{cases} 1 - \text{если } j = 2; \\ 0 - \text{если } j \neq 2; \end{cases} x_J = \begin{cases} 1 - \text{если } j = J; \\ 0 - \text{если } j \neq J. \end{cases} \quad (1)$$

Тогда  $f(j)=f(x_j)$  в РЛУ записывается в виде:

$$r = \begin{cases} 1 - \text{если } f(x_j) = 1; \\ 0 - \text{если } f(x_j) = 0. \end{cases} \quad (2)$$

С учетом введенных понятий формулируется определение объединенного алгоритма  $A_o$ .

**Определение.** Объединенным называется алгоритм  $A_o$ , отвечающий следующим условиям:

- любой элемент алфавита  $(O, Q, G, A) A_j, j=1, J$  входит в ПЛСА  $A_o$  и притом только один раз;
- если в функции  $f(x_j)$  ПЛСА  $A_o$ , включающий РЛУ, подставить конкретное значение  $j$ , то последовательность выполнения элементов алфавита  $O, Q, G, A$  будет соответствовать ПЛСА  $A_j$ .

Таким образом, задачей объединения АОЗ является синтез алгоритма  $A_o$ , отвечающего указанным условиям.

## 2 Метод объединения алгоритмов обслуживания заявок

Очевидно, что метод объединения АОЗ должен учитывать особенности алгоритмического отображения ССМО и его использования в качестве основы построения GPSS-модели:

- наличие параллельных ветвей и соответственно операторов их разделения и объединения;
- одновременное присутствие в модели множества заявок, находящихся на различных стадиях выполнения АОЗ;
- наличие в логических узлах  $A_o$  РЛУ, определяющих маршрут обслуживания заявки  $h_j$  с помощью анализа значений логических условий  $r=f(x_j)$ .

Очевидно, что проверки дополнительных логических условий приводят к усложнению модели. Поэтому в результате объединения АОЗ необходимо

получить алгоритм  $A_o$ , отвечающий вышеприведенному определению при минимальном количестве вхождений РЛУ.

Известные методы объединения алгоритмов (метод преобразования и минимизации системы скобочных формул перехода [20] и метод кодирования входных переменных микропрограммного автомата [19]) предназначены для решения задач синтеза микропрограммных автоматов и не учитывают указанных выше особенностей АОЗ.

Кодирование входных переменных в [19] производится в предположении неизменности состояния входа микропрограммного автомата во время выполнения алгоритма, заданного соответствующим входным набором, тогда как в модели может одновременно обрабатываться множество различных заявок. При этом параллельно должны выполняться различные АОЗ, кодируемые разными состояниями входных переменных. Количество входных переменных определяется по числу объединяемых алгоритмов и, исходя из этого, минимизируется вхождение переменных в объединенный алгоритм.

Для построения GPSS-модели, в связи с требованием сокращения затрат времени на проведение имитационного эксперимента (ИЭ), количество входных переменных или РЛУ должно определяться прежде всего из условия минимальности числа их вхождений. При этом допустима избыточность типов РЛУ, т. е. РЛУ, в которых проверяются различные функции  $f(x_j)$ .

На основе анализа определяется необходимость разработки метода объединения АОЗ, учитывающего изложенные выше требования.

Основой предлагаемого метода объединения алгоритмов является анализ свойств разбиений  $\Pi$  на множестве  $A$  ( $A_j \in A, j=1, J$ ) объединяемых алгоритмов, позволяющий определить количество, расстановку РЛУ  $r_m$  в  $A_o$  и состав функций  $f(x_j)$  для каждого  $r_m, m = \overline{1, M}$ .

Членами разбиения  $\Pi_{z_\beta}, \beta=1, B$  являются индексы типов заявок ( $j$ ) алгоритмов  $A_j, j=1, J$ , включающих элемент  $z_\beta$ . Для определения и анализа  $\Pi_{z_\beta}$  построим матрицу вхождений  $M[E, J]$ , где  $E$  – количество столбцов, а  $J$  – строк матрицы. Количество столбцов матрицы определяется по выражению  $E = \Xi + E_q + 2(E_g + E_\alpha)$ , где  $\Xi$  – количество операторов множества  $O$ ;  $E_q, E_g, E_\alpha$  – количество ждущих, собственно логических и вероятных условий используемых в описании  $A_o$ . Столбцы матрицы помечаются элементами описания  $A_o$ , причем для каждого вероятностного и собственно логического условия выделяются два столбца, один из которых предназначен для последователей  $\alpha_n, g_s$  при истинном значении, а другой – при ложном. Членами  $m_{e,j}, e=1, E, j=1, J$  матрицы  $M[E, J]$  являются элементы  $z_\beta, \beta=1, B$ . На пересечении  $j$ -й строки  $e$ -го столбца, помеченного элементом  $z_\beta$ , записывается последователь  $z_\beta$  в  $j$ -м алгоритме. Наличие параллельных ветвей  $A_j$  при построении  $M[E, J]$  учитывается следующим образом. Если последователем  $z_\beta$  в  $j$ -м алгоритме является оператор распараллеливания  $R^{\uparrow p}$ , то в качестве  $m_{e,j}$  записывается этот оператор и его последователи, т. е. начальные элементы соответствующих параллельных ветвей. Если последо-

вателем  $z_\beta$  в  $j$ -м алгоритме является оператор объединения  $\downarrow^p \&$ , то в качестве  $m_{e,j}$  записывается этот оператор и его последователь. Оператор  $\downarrow^p \&$  записывается для всех элементов  $z_\beta$ , являющихся конечными в объединяемых им параллельных ветвях  $j$ -го алгоритма. При наличии неодинаковых начальных элементов в АОЗ первый столбец матрицы вхождений помечается фиктивным оператором  $o^n$ , который условно принимается начальным для всех  $A_j, j=1, J$ . В случае, если оператор является конечным, в качестве его последователя записывается фиктивный конечный оператор  $o^k$ .

Матрица вхождений позволяет построить разбиения  $\Pi_{z_\beta}$  для всех  $z_\beta, \beta=\overline{1, B}$ , входящих в  $J$  объединяемых алгоритмов. Причем  $\Pi_{z_\beta} = \{a_1, a_2, \dots, a_\sigma, a_\chi\}$ , где  $a_1, a_2, \dots, a_\sigma, a_\chi$  – блоки разбиения. Для определения состава блоков достаточно проанализировать столбцы матрицы вхождений. Номера строк  $j$ , имеющих одинаковые элементы  $m_{e,j}$  в  $e$ -м столбце, составляют блок  $a_\sigma$  разбиения  $\Pi_{z_\beta}$  для  $z_\beta$ , которым помечен данный столбец. Множество блоков, получаемых по  $e$ -му столбцу, образуют разбиение  $\Pi_{z_\beta}$ . По разбиению  $\Pi_{z_\beta}$  однозначно определяется количество проверок, т. е. вхождений  $r_m$  в  $A_o$  после выполнения элемента  $z_\beta$ . Это количество равно  $\chi - 1$ , где  $\chi$  – число блоков в разбиении. Аналогично строятся разбиения по всем  $e=1, E$  столбцам матрицы вхождений. Очевидно, что количество блоков по всем полученным разбиениям однозначно определяет общее количество вхождений МРЛУ  $r_m$  в алгоритм  $A_o$ , которое является минимальным для данных  $J$ , объединяемых в АОЗ. Количество вхождений РЛУ в  $A_o$  определяется по следующему выражению:

$$M = \sum_{\xi} (s_b - 1) + \sum_k (s_q - 1) + \sum_s (s_g - 1) + \sum_s (s_{-g} - 1) + \sum_n (s_\alpha - 1) + \sum_n (s_{-\alpha} - 1), \tag{3}$$

где  $s_b, s_q, s_g, s_{-g}, s_\alpha, s_{-\alpha}$  – соответственно количество блоков нетривиальных разбиений [19] по всем  $z_\beta$  с учетом разбиений для элементов  $g, \alpha$  при их истинном ( $s_g, s_\alpha$ ) и ложном ( $s_{-g}, s_{-\alpha}$ ) значениях.

Анализ разбиений  $\Pi_{z_\beta}$  позволяет определить состав логических функций  $f(x_j)$ , проверяемых в каждом РЛУ  $r_m, m=1, M$ . Элементами функций  $f(x_j)$  являются переменные  $x_j$ , имеющие индекс  $j$  тех алгоритмов  $A_j$ , которые входят в один блок разбиения  $\Pi_{z_\beta}$ . Пусть, например,  $\Pi_{z_\beta} = \{1, 2, 3; 4, 5; 10; 12, 14\}$ , где  $\alpha_1=1, 2, 3, \alpha_2=4, 5, \alpha_3=10, \alpha_4=12, 14, \chi=4$ . Тогда, после реализации элемента  $z_\beta$ , необходимо проверить  $\chi - 1$ , т. е. три РЛУ –  $r_1, r_2, r_3$ , которые образуют дерево проверок (рис. 1). Функция  $f(x_j)$  в этом случае имеет следующий вид:

$$fr_1 = x_4 \vee x_5, \quad fr_2 = x_{10}, \quad fr_3 = x_{12} \vee x_{14}, \tag{4}$$

при этом

$$\begin{aligned}
 r_1 &= \begin{cases} 1 - \text{если } f_{r_1} = 1 (j = 4 \text{ или } 5); \\ 0 - \text{если } f_{r_1} = 0; \end{cases} \\
 r_2 &= \begin{cases} 1 - \text{если } f_{r_2} = 1 (j = 10); \\ 0 - \text{если } f_{r_2} = 0; \end{cases} \\
 r_3 &= \begin{cases} 1 - \text{если } f_{r_3} = 1 (j = 12 \text{ или } 14); \\ 0 - \text{если } f_{r_3} = 0. \end{cases}
 \end{aligned} \tag{5}$$

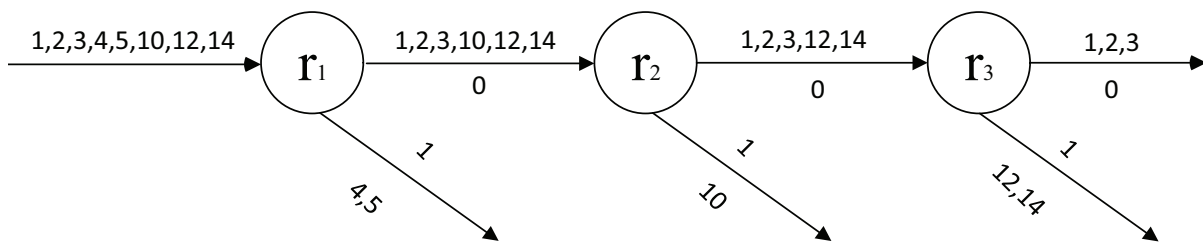


Рис. 1. Дерево проверок для РЛУ  $r_1, r_2, r_3$

Аналогичным образом определяются логические функции  $f(x_j)$  для всех нетривиальных разбиений  $\Pi_{z\beta}$ , полученных по матрице вхождений.

Далее проанализируем полученное множество функций  $f(x_j)$  с целью уменьшения их количества. Пусть имеются два двухблочных разбиения:  $\Pi_{z_1} = \{1, 2; 5, 7, 8\}$  и  $\Pi_{z_2} = \{1, 2; 7, 8, 9, 10\}$ . После выполнения  $z_1$  и  $z_2$  следует проверить по одному РЛУ –  $r_1$  и  $r_2$ . При этом в  $r_1$  могут проверяться две функции –  $f_{r_1} = x_1 \vee x_2$  или  $f_{r_1} = x_5 \vee x_7 \vee x_8$ . Аналогично для  $r_2$ :  $f_{r_2} = x_1 \vee x_2$  или  $f_{r_2} = x_7 \vee x_8 \vee x_9 \vee x_{10}$ . Очевидно, что в обоих случаях целесообразно использовать одну и ту же функцию  $f_r = x_1 \vee x_2$ . Аналогичные рассуждения могут быть справедливы для разбиений  $\Pi$ , имеющих большее количество блоков. Тогда для сокращения набора проверяемых функций по всем  $r_m, m = 1, M$  достаточно выявить в  $\Pi_{z\beta}, \beta = 1, B$  одинаковые блоки и использовать их для построения функций  $f(x_j)$ . Кроме того, сокращение количества  $f(x_j)$  достигается с помощью анализа не полностью определенных разбиений  $\Pi_{z\beta}$ , блоки которых содержат не все  $J$ -е номера объединенных алгоритмов.

Пусть имеются два разбиения  $\Pi_{z_4} = \{1, 2, 3, 4; 5, 6\}$  и  $\Pi_{z_5} = \{1, 4; 10, 12\}$ . После выполнения  $z_4$  и  $z_5$  проверяется по одному РЛУ –  $r_4$  и  $r_5$ .

В  $r_4$  могут проверяться две функции –  $f_{r_4} = x_1 \vee x_2 \vee x_3 \vee x_4$  или  $f_{r_4} = x_5 \vee x_6$ , а в  $r_5$  –  $f_{r_5} = x_1 \vee x_4$  или  $f_{r_5} = x_{10} \vee x_{12}$ . РЛУ  $r_4$  и  $r_5$  образуют вершины проверок, приведенные на рис. 2.

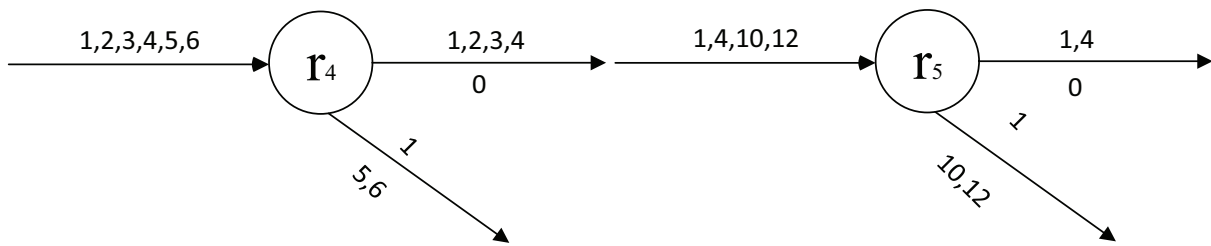


Рис. 2. Вершины проверок для РЛУ  $r_4, r_5$

Дополним разбиение  $\Pi_{z_5}$ , включив в первый блок переменные  $x_2$  и  $x_3$ . Сделать это позволяет то обстоятельство, что в узел  $r_5$  заявки с  $j = 2$  или  $j = 3$  не попадают, так как маршрут их обработки проходит по другим элементам  $z_\beta$ . Тогда в  $r_4$  и  $r_5$  достаточно проверить одну и ту же функцию  $f_r = x_1 \vee x_2 \vee x_3 \vee x_4$ . При этом  $r_5$  образует проверку вида, показанного на рис. 3.

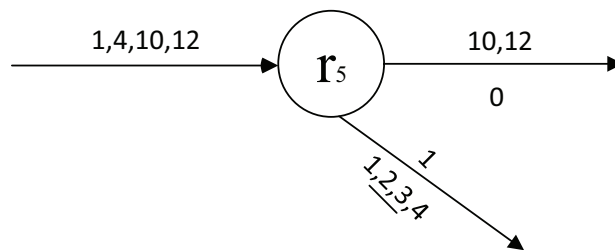


Рис. 3. Вершина проверок для РЛУ  $r_5$  после добавления переменных  $x_2$  и  $x_3$

Для сокращения набора проверяемых  $f(x_j)$  с учетом указанного положения достаточно проанализировать не полностью определенные разбиения по всем столбцам матрицы вхождений  $e = \overline{1, E}$ , выявить возможность дополнения блоков и использовать их для построения соответствующих функций.

Таким образом, полученные матрицы вхождений  $M [E, J]$ , разбиения  $\Pi$  по всем столбцам  $e=1, E$  и набор функций  $f(x_j)$  дают возможность построить объединенную ПЛСА  $A_o$  при однократном вхождении элементов  $z_\beta, \beta = \overline{1, B}$ , минимальном количестве  $M$  проверок  $r_m, m = \overline{1, M}$  и сокращенном наборе функций  $f(x_j)$ .

Для построения объединенной ПЛСА алгоритма  $A_o$  выписывается элемент  $z_1$ , которым помечен первый столбец матрицы  $M [E, J]$ . Таким элементом может быть фиктивный оператор  $o^n$  или начальный элемент, одинаковый для всех алгоритмов  $A_j, j = \overline{1, J}$ . По разбиению  $\Pi_{z_1}$  и набору функций  $f(x_j)$  определяются РЛУ  $r_m$  и соответствующие им функции  $f_{r_m}(x_j)$ . Полученные РЛУ записываются в ПЛСА  $A_o$  после элемента  $z_1$ . Нумерация РЛУ и стрелок осуществляется по мере их записи в ПЛСА. Далее, по матрице  $M [E, J]$  выписывается элемент, выполняемый после  $z_1$  при истинном значении полученных РЛУ. Для этого элемента выполняются операции определения и записи в ПЛСА соответствующих РЛУ и функций  $f(x_j)$ . Указанные процедуры повторяются до записи в ПЛСА конечного элемента  $o^k$ . Затем по матрице  $M [E, J]$  вписывается элемент, выполняющийся



после  $z_1$ , при ложном значении первого РЛУ. Если этот элемент был выписан на предыдущем шаге, то переход к нему осуществляется по стрелке, которая приписана данному РЛУ. Далее переходим к элементу, который выполняется при истинном значении первого и ложном второго РЛУ, и т. д. Запись членов ПЛСА продолжается до тех пор, пока не будут перебраны все элементы, которыми помечены столбцы матрицы  $M[E, J]$ , а также все РЛУ, полученные на предыдущих шагах при их ложном значении. Если последователем элемента является оператор распараллеливания  $R\uparrow^p$ , то он выписывается в ПЛСА со своим номером, а его стрелке ( $\uparrow^p$ ) присваивается очередной по порядку номер. Начальные члены параллельных ветвей, образуемых  $R\uparrow^p$ , выписываются в процессе построения  $A_o$  последовательно и перед каждым ставится стрелка с тем же значением индекса  $p$ . Для каждого начального члена параллельной ветви осуществляются процедуры определения последователей и РЛУ, аналогично описанным выше. После элемента, являющегося последним, в параллельной ветви, объединяемой оператором  $\downarrow^p\&$  (такой элемент в матрице  $M[E, J]$  помечается символом  $\&$ ), в ПЛСА при соответствующих значениях РЛУ записывается тождественно ложное логическое условие  $w\uparrow^p$ , заканчивающее параллельную ветвь. Если последователем элемента является оператор объединения  $\downarrow^p\&$ , то в ПЛСА записывается этот оператор и его последователь, для которого также повторяются указанные ранее процедуры. Оператор объединения помечается стрелкой  $\downarrow^p$  с тем же значением индекса  $p$ , которые имеют стрелки тождественно ложных логических условий  $w\uparrow^p$ , заканчивающих соответствующие параллельные ветви.

В результате указанных процедур синтезируется объединенная ПЛСА  $A_o$  при однократном вхождении элементов  $z_\beta$ , минимальном количестве проверок  $r_m$ ,  $m=1, M$  и сокращенном наборе функций  $f(x_j)$ .

## Заключение

Жизненный цикл сложных систем [21], к которым относятся СОЖЦ ЖАТ [22], включает следующие основные стадии:

- формулировка проблемы;
- определение целей исследования;
- выбор парадигмы моделирования;
- постановка задач моделирования;
- синтез концептуальной модели и формализованной схемы исследуемой системы;
- выбор инструментальных средств построения имитационной модели (ИМ);
- определение требований к ИМ;
- разработка моделирующей программы;
- планирование и выполнение контрольных экспериментов;

- проведение экспертизы выполнения всех стадий разработки и данных контрольных экспериментов – верификация ИМ [23];
- заключение по результатам верификации ИМ;
- оформление программной документации;
- архивирование моделирующей программы в соответствующих фондах алгоритмов и программ.

Проведение исследований с использованием разработанной ИМ предполагает реализацию процедур ее валидации для конкретных приложений и выполнение указанных выше стадий жизненного цикла ИМ. Следует отметить, что метод имитационного моделирования не структурирован, что затрудняет как разработку, так и выполнение процедур верификации ИМ СОЖЦ ЖАТ и валидации приложений. В связи с этим актуальной является задача структуризации ИМ в конкретной предметной области – СОЖЦ ЖАТ. Использование в качестве базы алгоритмического отображения СОЖЦ ЖАТ позволяет структурировать ИМ за счет формализации АОЗ на языке ПЛСА, что обеспечивает возможности контроля полноты и корректности алгоритмов, а также их формальных преобразований. В данной работе рассмотрены задачи и методы формализации и формальных преобразований алгоритмического отображения СОЖЦ ЖАТ.

Полученные результаты:

- сформулированы требования к алгоритмическому описанию СОЖЦ ЖАТ как ССМО;
- формализованное описание АОЗ предложено выполнять на языке ПЛСА, обеспечивающего возможность сокращения общего количества элементов описания алгоритмов;
- разработан метод алгоритмического описания ССМО на основе анализа разбиений по матрице вхождений элементов ПЛСА в АОЗ, позволяющий минимизировать количество разделительных логических условий, логических функций при однократном вхождении элементов исходного алфавита;
- предложенный метод объединения АОЗ позволяет сократить объем алгоритмического отображения ССМО, а следовательно, объем GPSS-модели и количество ошибок в ней.

Перспективы:

- разработка алгоритма синтеза объединенного АОЗ;
- разработка методики оценки эффективности объединения АОЗ на основе анализа количества вхождений элементов исходного алфавита ПЛСА в  $A_j$  и суммарного количества вхождений элементов исходного алфавита ПЛСА и РЛУ в объединенный алгоритм;
- разработка компьютерной программы объединения АОЗ на основе предложенного метода с принятием решений по критерию минимального количества вхождений элементов исходного алфавита и РЛУ в алгоритмическое описание ССМО.

## Библиографический список

1. Сапожников Вл. В. Доказательство безопасности систем ж.-д. автоматики / Вл. В. Сапожников, О. А. Наседкин // Наука и транспорт. – 2006. – С. 10–13.
2. Наседкин О. А. Особенности испытания МПУ ЖАТ / О. А. Наседкин, Е. В. Ледеяев // Автоматика связь, информатика. – 2012. – № 7. – С. 30–32.
3. Васильев Д. А. Экспертная программа для проведения испытаний технологического программного обеспечения систем микропроцессорной централизации / Д. А. Васильев, С. В. Гизлер, О. А. Наседкин, М. П. Шайфер // Развитие элементной базы и совершенствование методов построения устройств железнодорожной автоматики и телемеханики : сб. науч. тр. ; под. ред. Вл. В. Сапожникова. – СПб. : ФГБОУ ВПО ПГУПС, 2014. – С. 39–42.
4. Насонов Г. Ф. Автоматизированная система мониторинга проектирования, производства, строительства и проведения пусконаладочных работ по системам СЦБ / Г. Ф. Насонов, М. Н. Василенко, П. Е. Булавский // Транспорт Российской Федерации. – 2010. – № 3. – С. 46–49.
5. Василенко М. Н. Автоматизированная система экспертизы схемных решений железнодорожной автоматики и телемеханики / М. Н. Василенко, А. М. Горбачев, Д. В. Зуев, Е. В. Григорьев // Транспорт Российской Федерации. – 2011. – № 5. – С. 64–67.
6. Молодцов В. П. Системы диспетчерского контроля и мониторинга устройств железнодорожной автоматики и телемеханики : учеб. пособие / В. П. Молодцов, А. А. Иванов. – СПб. : Петербургский гос. ун-т путей сообщения, 2010. – 140 с.
7. Ефанов Д. В. Обеспечение безопасности движения за счет технического диагностирования и мониторинга устройств железнодорожной автоматики и телемеханики / Д. В. Ефанов, П. А. Плеханов // Транспорт Урала. – 2011. – № 3. – С. 44–48.
8. Лыков А. А. Техническое диагностирование и мониторинг состояния устройств ЖАТ / А. А. Лыков, Д. В. Ефанов, С. В. Власенко // Транспорт Российской Федерации. – 2012. – № 5. – С. 67–72.
9. Ефанов Д. В. Некоторые аспекты развития систем функционального контроля устройств железнодорожной автоматики и телемеханики / Д. В. Ефанов // Транспорт Урала. – 2015. – № 1. – С. 35–40.
10. Тележенко Т. А. Автоматизированная система экспертизы схемных решений ЖАТ / Т. А. Тележенко // Автоматика, связь, информатика. – 2009. – № 5. – С. 24–26.
11. Василенко М. Н. Электронный документооборот в хозяйстве СЦБ / М. Н. Василенко, В. Г. Трохов, Д. В. Зуев // Автоматика, связь, информатика. – 2014. – № 8. – С. 2–3.
12. Василенко М. Н. Развитие электронного документооборота в хозяйстве АТ / М. Н. Василенко, В. Г. Трохов, Д. В. Зуев, Д. В. Седых // Автоматика, связь, информатика. – 2015. – № 1. – С. 14–16.
13. Нестеров В. В. Развитие систем СТДМ, АСУ-Ш-2 и АОС-ШЧ / В. В. Нестеров // Автоматика, связь, информатика. – 2012. – № 12. – С. 45–46.
14. Информационные технологии : учебник для вузов / Б. Я. Советов, В. В. Цехановский. – М. : Высшая школа, 2003. – 263 с.

15. Булавский П. Е. Матричный метод формализации имитационных моделей сложных систем массового обслуживания / П. Е. Булавский, Д. С. Марков // Известия Петербургского университета путей сообщения. – 2010. – № 4. – С. 186–195.
16. Алиев Т. И. Основы моделирования дискретных систем / Т. И. Алиев. – СПб. : СПбГУ ИТМО, 2009. – 363 с.
17. GPSSWORLD. Основы имитационного моделирования различных систем / Е. М. Кудрявцев. – М. : ДМК Пресс, 2004. – 317 с.
18. Марков Д. С. Метод порционного моделирования транспортных систем массового обслуживания / Д. С. Марков, В. Б. Соколов // Развитие элементной базы и совершенствование методов построения устройств железнодорожной автоматики и телемеханики : сб. науч. тр. ; под ред. Вл. В. Сапожникова. – СПб. : ФГБОУ ВПО ПГУПС, 2014. – С. 43–47.
19. Синтез управляющих автоматов / В. Г. Лазарев, Е. И. Пийль. – М. : Энергия, 1984. – 408 с.
20. Дьяченко В. Ф. Управление на сетях связи / В. Ф. Дьяченко, В. Г. Лазарев, Г. Г. Саввин. – М. : Наука, 1967. – 223 с.
21. Balci O. Verification, validation and accreditation / O. Balci // Proceedings of the 1998 Winter Simulation Conference, Washington, DC, USA, December 13–16, Piscataway, NJ, 1998. – Pp. 41–48.
22. Шаманов В. И. Управление процессом модернизации комплексов систем железнодорожной автоматики и телемеханики / В. И. Шаманов // Автоматика на транспорте. – 2015. – Т. 1. – № 3. – С. 237–250.
23. Carson J. S. Model verification and validation / J. S. Carson // Proceedings of the 2002 Winter Simulation Conference, San Diego, CA, USA, December 08–11, 2002. – Pp. 52–58.

*Bulavskij Petr E.,*

*Markov Dmitry S.,*

*Sokolov Vadim B.,*

*Konstantinova Tatjana Yu.*

«Automation and Remote Control on Railways» department  
Petersburg state transport university

### **Formalisation of algorithmic description of systems of railway automation and remote control life cycle provision**

This paper is devoted to the problem of synthesis of algorithmic reflection of railway automation and remote control life cycle as complex queuing system (CQS). It is accepted that algorithmic reflection is presented as request service algorithm (RSA), it should be formalised on PLSA language and used as the base for simulation models (SM) of CQS. Expediency of synthesis of united RSA is shown. The definition of united RSA is given. Authors offer the method of algorithm union based on formation of occurrence matrix

of basic PLSA alphabet, analysis of its fragmentation, that allows to synthesize the united RSA with occurring once of basic PLSA alphabet elements, minimum number of separation logical conditions and minimized number of dividing logical conditions logical functions. Implementation of offered method of alorythm union allows to reduce the amount of SM and therefore the number of errors in simulation software.

railway automation and remote control life cycle provision systems; complex queuing systems; simulation modeling; request service alorythm; united alorythm; occurrence matrix; separation logical conditions

### Reference

1. Sapozhnikov V.I., Nasedkin O.A. Safety proof of railway automation systems. *Science & Transport*, 2006, pp. 10–13.
2. Nasedkin O.A., Lediaev E.V. Features of test of computer based railway automation and remote control devices. *Automatics, Communication, Informatics*, 2012, № 7, pp. 30–32.
3. Vasiljev D.A., Gizler S.V., Nasedkin O.A., Shaipher M.P. Expert software for testing technology software of computer based interlocking. Component base development and railway automation and remote control devices design methods improvement; eds. V.I. Sapozhnikov. St. Petersburg, PSTU, 2014, pp. 39–42.
4. Nasonov G.F., Vasilenko M.N., Bulavsky P.E. Automated system for the monitoring of railway signaling systems design, manufacturing, construction and commissioning. *Transport RF*, 2010, № 3, pp. 46–49.
5. Vasilenko M.N., Gorbacev A.M., Zuev D.V., Grigorjev E.V. Automated system for the examination of railway automation and remote control schematics. *Transport RF*, 2011, № 5, pp. 64–67.
6. Molodtsov V.P., Ivanov A.A. Supervisory and monitoring systems of railway automation and remote control devices. St. Petersburg, PSTU, 2010, 140 p.
7. Efanov D.V., Plekhanov P.A. Traffic safety provision by technical diagnostics and monitoring of railway automation and remote control devices. *Ural Transport*, 2011, № 3, pp. 44–48.
8. Lykov A.A., Efanov D.V., Vlasenko S.V. Technical diagnostics and monitoring of railway automation and remote control. *Transport RF*, 2012, № 5, pp. 67–72.
9. Efanov D.V. Some aspects of concurrent error detection systems of railway automation and remote control devices development. *Ural Transport*, 2015, № 1, pp. 35–40.
10. Telezhenko T.A. Automated system for the examination of railway automation and remote control schematics. *Automatics, Communication, Informatics*, 2009, № 5, pp. 24–26.
11. Vasilenko M.N., Trohov V.G., Zuev D.V. Electronic document management in railway signaling. *Automatics, Communication, Informatics*, 2014, № 8, pp. 2–3.
12. Vasilenko M.N., Trohov V.G., Zuev D.V., Sedych D.V. Development of electronic document management in railway automation and remote control. *Automatics, Communication, Informatics*, 2015, № 1, pp. 14–16.

13. Nesterov V. V. Development of STDM, ASU-Sch-2 and AOS-SchCh systems. *Automatics, Communication, Informatics*, 2012, № 12, pp. 45–46.
14. Sovetov B. Ya., Tsehanovsky V. V. *Information technology*. Moscow, High School, 2003, 263 p.
15. Bulavsky P. E., Markov D. S. The matrix method of complex queuing system simulation models formalization. *Proceedings of Petersburg Transport University*, 2010, № 4, pp. 186–195.
16. Aliev T. I. *Basics of digital systems modeling*. St. Petersburg, ITMO University, 2009, 363 p.
17. Kudryavtsev E. M. *GPSSWORLD. Basics of different systems modeling*. Moscow, DMK Press, 2004, 317 p.
18. Markov D. S., Sokolov V. B. The method of portion modeling of transport queuing systems. *Component base development and railway automation and remote control devices design methods improvement*; eds. Vl. V. Sapozhnikov. St. Petersburg, PSTU, 2014, pp. 43–47.
19. Lazarev V. G., Pijil E. I. *Synthesis of regulatory finite state machines*. Moscow, Energia, 1984, 408 p.
20. Djachenko V. F., Lazarev V. G., Savvin G. G. *Control in communication network*. Moscow, Nauka, 1967, 223 p.
21. Balci O. Verification, validation and accreditation. *Proceedings of the 1998 Winter Simulation Conference*, Washington, DC, USA, December 13–16, Piscataway, N. J., 1998, pp. 41–48.
22. Shamanov V. I. Control of railway automation and remote control system complexes modernization process. *Automation on Transport*, 2015, vol. 1, № 3, pp. 237–250.
23. Carson J. S. Model verification and validation. *Proceedings of the 2002 Winter Simulation Conference*, San Diego, CA, USA, December 08–11, 2002, pp. 52–58.

*Представлена к публикации членом редколлегии Вал. В. Сапожниковым  
Поступила в редакцию 01.09.2015, принята к публикации 24.09.2015*

БУЛАВСКИЙ Петр Евгеньевич – доктор технических наук, профессор кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I.  
e-mail: pbulavsky@gmail.com

МАРКОВ Дмитрий Спиридонович – кандидат технических наук, доцент кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I.  
e-mail: MDS1945@yandex.ru

СОКОЛОВ Вадим Борисович – кандидат технических наук, доцент кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I.

e-mail: SVB9@yandex.ru

КОНСТАНТИНОВА Татьяна Юрьевна – кандидат технических наук, доцент кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения Императора Александра I.

e-mail: at-tanya@mail.ru

© Булавский П. Е., 2015

© Марков Д. С., 2015

© Соколов В. Б., 2015

© Константинова Т. Ю., 2015