Автоматизация и моделирование в проектировании и управлении. 2023. № 4 (22). С. 4-11.
ISSN 2658-3488 print, ISSN 2658-6436 online
Automation and modeling in design and management. 2023. № 4 (22). P. 4-11.

Автоматизация и управление технологическими процессами и производствами, системы автоматизации проектирования

Научная статья
Статья в открытом доступе
УДК 331.101.01
doi: 10.30987/2658-6436-2023-4-4-11

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ АВТОМАТНОГО ПРОГРАММИРОВАНИЯ ДЛЯ РАЗРАБОТКИ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ПУНКТОМ ПРОПУСКА ЛЮДЕЙ И АВТОТРАНСПОРТА С ОДНИМ ВЪЕЗДОМ И ОДНИМ ВЫЕЗДОМ

Павел Васильевич Гумунюк[1]✉, Александр Николаевич Шурпо[2]✉
[1, 2] Институт конструкторско-технологической информатики Российской академии наук, г. Москва, Россия
[1] pavel05091997@yandex.ru
[2] a-shurpo@yandex.ru, https://orcid.org/0000-0003-1962-1969

Аннотация. *Проведён анализ пропускной системы, которую можно устанавливать и использовать для предприятий, зданий и территорий, с целью выяснения – насколько сложна разработка таких систем управления (шлагбаумами, воротами и другими средствами контроля и безопасности), основанных на моделях автоматного программирования и на инструментах для их реализации. Система имеет два шлагбаума (предназначены для въезда и выезда автомобилей с территории) и один проход для людей. Такие пропускные системы достаточно часто применяются, но, в подавляющем большинстве, данные средства контроля и безопасности редко автоматизированы. Автоматизация, за счёт наличия умных устройств, позволит снизить время, затрачиваемое на пропуск автомобилей и людей, оставляя за человеком лишь роль наблюдателя с правом вмешательства в процесс при необходимости. При построении системы управления, на основе автоматного программирования необходимо: во-первых, рассмотреть технологический процесс пропускного пункта, во-вторых, построить логику на основании выделения явных состояний автоматической системы. При должном рассмотрении и детальном изучении технологического процесса, с построением логики, не должно возникнуть проблем и код программы будет работать безотказно, так как он учитывает все возможные варианты поведения автоматизированной системы.*

Ключевые слова: пропускная система, умный контроль, switch – технология, автоматное программирование, система управления, индустрия 4.0.

Для цитирования: Гумунюк П.В., Шурпо А.Н. Применение технологии автоматного программирования для разработки системы автоматического управления пунктом пропуска людей и автотранспорта с одним въездом и одним выездом // Автоматизация и моделирование в проектировании и управлении. 2023. №4 (22). С. 4-11. doi: 10.30987/2658-6436-2023-4-4-11.

Original article
Open Access Article

APPLYING AUTOMATA-BASED PROGRAMMING TECHNOLOGY TO DEVELOP AN AUTOMATIC CONTROL SYSTEM OF A CHECKPOINT FOR PEOPLE AND VEHICLES WITH ONE ENTRY AND ONE EXIT

Pavel V. Gumunyuk[1]✉, Aleksandr N. Shurpo[2]✉
[1, 2] Institute for Design-Technological Informatics of the Russian Academy of Sciences, Moscow, Russia

[1] pavel05091997@yandex.ru
[2] a-shurpo@yandex.ru, https://orcid.org/0000-0003-1962-1969

Abstract. *An analysis of the access system, which can be installed and used for enterprises, buildings and territories, is carried out to find out how difficult it is to develop such control systems (barriers, gates and other control and security means) based on automatic programming models and tools for their implementation. The system has two barriers (designed for vehicles entering and exiting the territory) and one passage for people. Such access systems are used quite often, but, in the overwhelming majority, these control and security measures are rarely automated. Automation, due to the presence of smart devices, will reduce the time spent on vehicles and people's passing, leaving the individual only to be an observer, with the right to intervene if it is necessary. When building a control system based on automatic programming, firstly, it is necessary to consider the technological process of the checkpoint, and secondly, to construct logic based on identifying the obvious states of the automatic system. With proper examination and detailed study of the technological process, there should be no problems with the construction of logic and the program code will work flawlessly, since it takes into account all possible behaviour options of the automated system.*

Keywords: access system, smart control, switch technology, automatic programming, control system, industry 4.0.

For citation: Gumunyuk P.V., Shurpo A.N. Applying Automata-Based Programming Technology to Develop an Automatic Control System of a Check-point for People and Vehicles With One Entry and One Exit. Automation and modeling in design and management, 2023, no. 4 (22). pp. 4-11. doi: 10.30987/2658-6436-2023-4-4-11.

Введение

В работе рассмотрен процесс написания программы для автоматического управления пропускной системой. Эту систему можно успешно применять как на предприятиях, так и на частных охраняемых территориях, поскольку она существенно увеличивает пропускную способность контрольных пунктов для людей и автотранспорта [1].

Материалы, модели, эксперименты и методы

Рассмотрим пропускную систему из двух шлагбаумов, которые установлены напротив друг друга и перекрывают один участок дороги [2].

Дорога соответственно разделена на две полосы: для движения на въезд и выезд. Между шлагбаумами установлен столб, разделяющий две полосы для движения транспорта, а рядом с полосами находится дверь для пропуска людей [3]. В стойках оснований шлагбаумов установлены датчики движения для предотвращения закрывания шлагбаумов, если под ними находится какой-либо объект (человек или автомобиль). На подъездах к шлагбаумам расположены стойки, необходимые для прикладывания пропусков водителями [4]. Внешний вид данного пропускного пункта представлен на рис. 1.

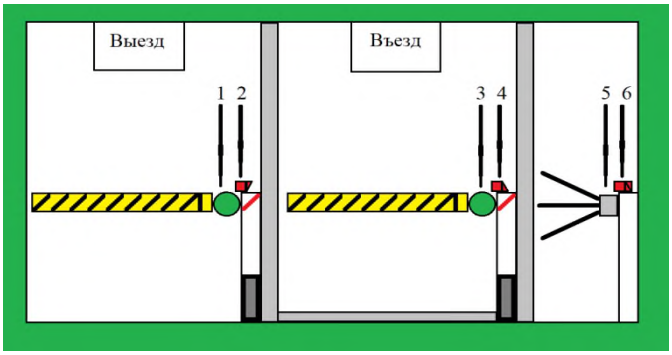

Рис. 1. Схематическое изображение пропускного пункта с одним въездом и одним выездом:
1 – пропуск на выезд; *2* – считыватель пропуска на выезд для автомобилей; *3* – пропуск на въезд;
4 – считыватель пропуска на въезд для автомобилей; *5* – проход для пешеходов;
6 – считыватель пропуска для посетителей

Fig. 1. Schematic representation of a control point with one entry and one exit
1 – exit pass; *2* – car exit pass reader; *3* – entry pass; *4* – an entry pass reader for cars;
5 – walkway for pedestrians; *6* – pass reader for visitors

Программа написана на высокоуровневом языке программирования C#, далее рассмотрим основные фрагменты кода [5].

Рассмотрим, сколько всего состояний может быть для данной системы [6]:

– шлагбаумы и двери закрыты;

– шлагбаум на въезд открыт – осуществляется пропуск автомобиля, шлагбаум на выезд закрыт и двери для прохода тоже;

– шлагбаум на въезд открыт, шлагбаум на выезд открыт – осуществляется одновременный въезд и выезд автомобилей, двери для прохода закрыты;

– оба шлагбаума и двери для прохода открыты – осуществляется пропуск по всем путям сразу;

– шлагбаум на въезд закрыт, шлагбаум на выезд открыт, двери для прохода закрыты – осуществляется только выезд автомобиля;

– оба шлагбаума закрыты, а двери для прохода открыты – осуществляется только вход и выход посетителей;

– шлагбаум на въезд закрыт, шлагбаум на выезд открыт, двери для прохода открыты – осуществляется выезд автомобиля и пропуск посетителей;

– шлагбаум на въезд открыт, шлагбаум на выезд закрыт, двери для прохода открыты – осуществляется выезд автомобиля и пропуск посетителей.

Структурно программа будет представлять следующую конструкцию: ядром логики будет являться оператор множественного выбора switch, который будет переключать состояния в зависимости от условий. Условий в коде восемь: «0: шлагбаумы и двери закрыты»; «1: шлагбаум на въезд открыт» – осуществляется пропуск автомобиля, шлагбаум на выезд закрыт и двери тоже; «2: шлагбаум на въезд открыт, шлагбаум на выезд открыт» – осуществляется одновременный въезд и выезд автомобилей, двери закрыты; «3: оба шлагбаума и двери открыты» – осуществляется пропуск по всем путям сразу; «4: шлагбаум на въезд закрыт, шлагбаум на выезд открыт, двери закрыты» – осуществляется только выезд автомобиля; «5: оба шлагбаума закрыты, а двери открыты» – осуществляется только вход и выход посетителей; «6: шлагбаум на въезд закрыт, шлагбаум на выезд открыт, двери открыты» – осуществляется выезд автомобиля и пропуск посетителей; «7: шлагбаум на въезд открыт, шлагбаум на выезд закрыт, двери открыты» – осуществляется выезд автомобиля и пропуск посетителей. Программная логика обладает следующими условиями: «все закрыто», «въезд машины», «выезд машины», «въезд и выезд машины», «въезд и выезд машины и проход посетителей», «проход посетителей», «въезд машины и проход посетителей», «выезд машины и проход посетителей». Из каждого состояния возможен переход в другое при выполнении условии этого состояния. Выполнение условий состояния осуществляется при помощи срабатывания датчиков системы автоматического управления.

Условие «все закрыто» достигается при закрытии всех дверей. Об этом программа может узнать при наличии следующих сигналов от автоматической системы: если imputControl == false пропуск на въезд автомобиля не получен и outputControl == false пропуск на выезд автомобиля не получен и при этом doorInControl == false пропуск на вход посетителя не получен и doorOutControl == false пропуск на выход посетителя не получен.

Условие «въезд машины» достигается при закрытии всех дверей, кроме двери на въезд машины. Об этом программа может узнать при наличии следующих сигналов от автоматической системы: если imputControl == true пропуск на въезд автомобиля получен и outputControl == false пропуск на выезд автомобиля не получен и при этом doorInControl == false пропуск на вход посетителя не получен и doorOutControl == false пропуск на выход посетителя не получен.

Условие «въезд и выезд машины» достигается при закрытии дверей для пеших посетителей и открытии дверей на въезд и выезд машины. Об этом программа может узнать при наличии следующих сигналов от автоматической системы: если imputControl == true пропуск на въезд автомобиля получен и outputControl == true пропуск на выезд автомобиля

получен и при этом doorInControl == false пропуск на вход посетителя не получен и doorOutControl == false пропуск на выход посетителя не получен.

Условие «проезд машин и проход посетителей» достигается при открытии всех дверей. Об этом программа может узнать при наличии следующих сигналов от автоматической системы: если imputControl == true пропуск на въезд автомобиля получен и outputControl == true пропуск на выезд автомобиля получен и при этом doorInControl == true пропуск на вход посетителя получен или doorOutControl == true пропуск на выход посетителя получен.

Условие «выезд машины» достигается при закрытии всех дверей, кроме двери на выезд машины. Об этом программа может узнать при наличии следующих сигналов от автоматической системы: если imputControl == false пропуск на въезд автомобиля не получен и outputControl == true пропуск на выезд автомобиля получен и при этом doorInControl == false пропуск на вход посетителя не получен и doorOutControl == false пропуск на выход посетителя не получен.

Условие «проход посетителей» достигается при закрытии всех дверей, кроме двери на вход и выход пеших посетителей. Об этом программа может узнать при наличии следующих сигналов от автоматической системы: если imputControl == false пропуск на въезд автомобиля не получен и outputControl == false пропуск на выезд автомобиля не получен и при этом doorInControl == true пропуск на вход посетителя получен или doorOutControl == true пропуск на выход посетителя получен.

Условие «выезд машины и проход посетителей» достигается при закрытии дверей для возможности въезда, открытии двери для возможности выезда и двери на вход и выход пеших посетителей. Об этом программа может узнать при наличии следующих сигналов от автоматической системы: если imputControl == false пропуск на въезд автомобиля не получен и outputControl == true пропуск на выезд автомобиля получен и при этом doorInControl == true пропуск на вход посетителя получен или doorOutControl == true пропуск на выход посетителя получен.

Условие «въезд машины и проход посетителей» достигается при закрытии дверей для возможности выезда, открытии двери для возможности въезда и двери на вход и выход пеших посетителей. Об этом программа может узнать при наличии следующих сигналов от автоматической системы: если imputControl == true пропуск на въезд автомобиля получен и outputControl == false пропуск на выезд автомобиля не получен и при этом doorInControl == true пропуск на вход посетителя получен или doorOutControl == true пропуск на выход посетителя получен.

Для упрощения программы как в плане логики, так и в плане структуры, в код была введена функция goToCase, которая выступает как универсальное решение для перечисления всех условий в собственном теле. Данная функция вызывается внутри каждого условия по ссылке, принимает в себя входные параметры, не запрашивая их в каждом условии и выдает команду по переходу к требуемому состоянию при наличии необходимых условий.

В коде программы присутсвует оператор do while и конструкция:

```
do
{
 gateInOpen = true; ( или gateOutOpen = true; )
}
 while ( inNoEmpty == false ) ( или while ( outNoEmpty == false ) )
```

Она необходима для предотвращения закрытия дверей при наличии людей или автомобилей в дверном проеме или шлагбауме, так как запрещает закрытие дверей на вход gateInOpen = true или на выход gateOutOpen = true пока на входе кто-то есть inNoEmpty == false или на выходе outNoEmpty == false.

Каждое состояние диктует свои команды исполнительным усутройствам.

Состояние 0: шлагбаумы и двери закрыты выдает директивы gateInOpen = false команда закрыть въездной шлагбаум gateOutOpen = false команда закрыть выездной шлагбаум doorOpen = false команда закрыть двери для посетителей.

Состояние 1: шлагбаум на въезд открыт выдает директивы gateInOpen = true команда открыть въездной шлагбаум gateOutOpen = false команда закрыть выездной шлагбаум doorOpen = false команда закрыть двери для посетителей.

Состояние 2: шлагбаум на въезд открыт, шлагбаум на выезд открыт выдает директивы gateInOpen = true команда открыть въездной шлагбаум gateOutOpen = true команда открыть выездной шлагбаум doorOpen = false команда закрыть двери для посетителей.

Состояние 3: оба шлагбаума и двери открыты выдает директивы gateInOpen = true команда открыть въездной шлагбаум gateOutOpen = true команда открыть выездной шлагбаум doorOpen = true команда открыть двери для посетителей.

Состояние 4: шлагбаум на въезд закрыт, шлагбаум на выезд открыт, двери закрыты выдает директивы gateInOpen = false команда закрыть въездной шлагбаум gateOutOpen = true команда открыть выездной шлагбаум doorOpen = false команда закрыть двери для посетителей.

Состояние 5: оба шлагбаума закрыты и двери открыты выдает директивы gateInOpen = false команда закрыть въездной шлагбаум gateOutOpen = false команда закрыть выездной шлагбаум doorOpen = true команда открыть двери для посетителей.

Состояние 6: шлагбаум на въезд закрыт, шлагбаум на выезд открыт, двери открыты выдает директивы gateInOpen = false команда закрыть въездной шлагбаум gateOutOpen = true команда открыть выездной шлагбаум doorOpen = true команда открыть двери для посетителей.

Состояние 7: шлагбаум на въезд открыт, шлагбаум на выезд закрыт, двери открыты выдает директивы gateInOpen = true команда открыть въездной шлагбаум gateOutOpen = false команда закрыть выездной шлагбаум doorOpen = true команда открыть двери для посетителей.

Ниже по тексту для наглядности представлен фрагмент кода с функцией goToCase и основной частью программы описывающий первое состояние автоматизированной системы управления.

С текстами программы можно ознакомиться в репозитории https://github.com/pavel–proger/AutomaticControl.

Определимся с переменными:

```
bool imputControl; // пропуск на въезд автомобиля получен.
bool outputControl; // пропуск на выезд автомобиля получен.
bool doorInControl; // пропуск на вход посетителя получен.
bool doorOutControl; // пропуск на выход посетителя получен.
bool inNoEmpty; // перед въездным шлагбаумом находится посторонний объект.
bool outNoEmpty; // перед выездным шлагбаумом находится посторонний объект.
bool gateInOpen; // команда открыть въездной шлагбаум, где true – открыть, а false – закрыть.
bool gateOutOpen; // команда открыть выездной шлагбаум, где true – открыть, а false – закрыть.
bool doorOpen; // команда открыть двери для прохода посетителей, где true – открыть, а false – закрыть.
int gateControl; // состояние пропускной системы.
void goToCase ( )
{ if ( imputControl == false and outputControl == false and (doorInControl == false or doorOutControl == false) )
{
 gateControl = 0;
}
else if ( imputControl == true and outputControl == false and (doorInControl == false or doorOutControl == false) )
{
 gateControl = 1;
}
```

```
else if ( imputControl == true and outputControl == true and (doorInControl == false or doorOutControl == false) )
{
 gateControl = 2;
}
else if ( imputControl == true and outputControl == true and (doorInControl == true or doorOutControl == true) )
{
 gateControl = 3;
}
else if ( imputControl == false and outputControl == true and (doorInControl == true or doorOutControl == true) )
{
 gateControl = 4;
}
else if ( imputControl == false and outputControl == false and (doorInControl == true or doorOutControl == true) )
{
 gateControl = 5;
}
else if ( imputControl == false and outputControl == true and (doorInControl == true or doorOutControl == true) )
{
 gateControl = 6;
}
else if ( imputControl == true and outputControl == false and (doorInControl == true or doorOutControl == true) )
{
 gateControl = 7;
}
}
switch ( gateControl )
{
case 0: // шлагбаумы и двери для прохода закрыты.
gateInOpen = false;
gateOutOpen = false;
doorOpen = false;
goToCase ( );
break;
}
```

Результаты

В данном случае нельзя было обойтись четырьмя состояниями. Одно, когда всё закрыто и по одному открытому состоянию на каждое из пропускных устройств, потому что, в таком случае, при одном открытом устройстве возникают проблемы с регулированием других устройств, так как каждое определённое состояние системы управления требует в данном случае чёткого контроля всех устройств системы. Из-за этого код программы станет больше и сложнее, но при этом система будет работать надежнее [7]. Switch – технология является ещё одним названием технологии автоматного программирования. Это связано с тем, что данная методика использует оператора множественного выбора switch для выбора устойчивого состояния (в коде выше можно это увидеть) [8]. Программный код в виде блок-схемы представлен на рис. 2.

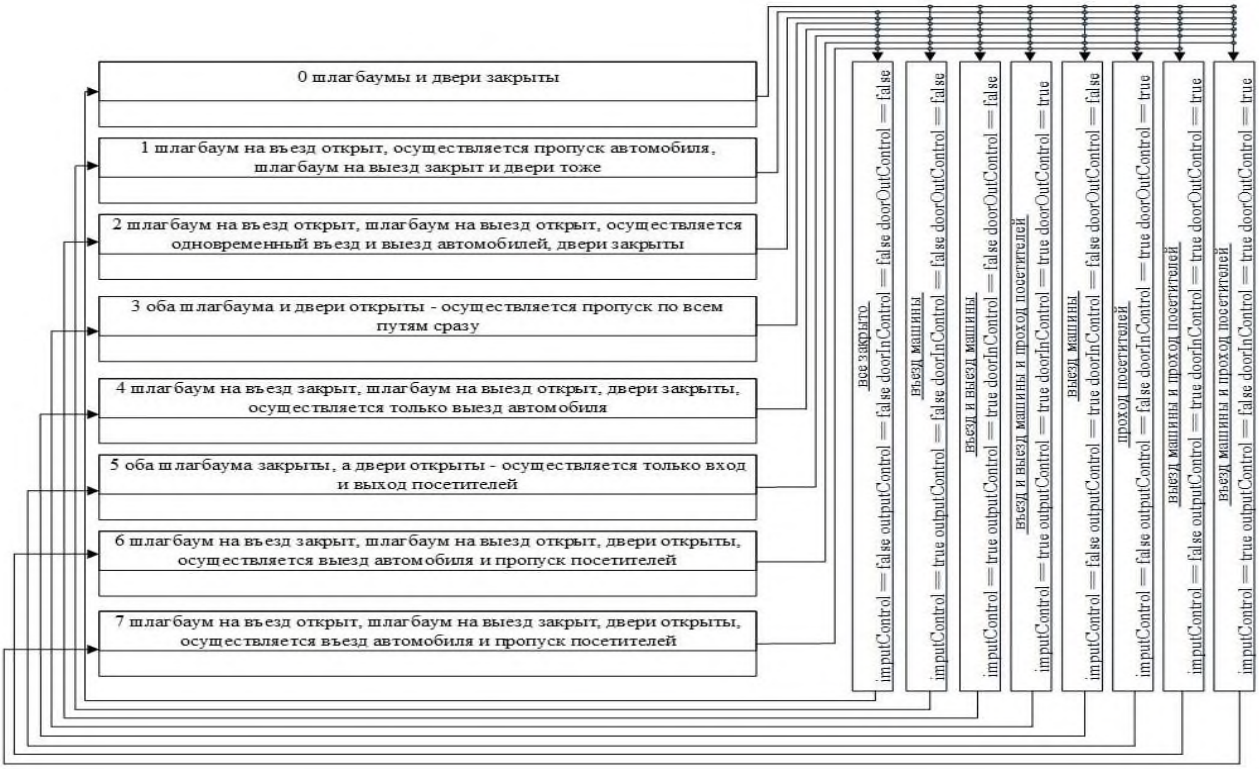

Рис. 2. Представление программного кода ввиде алгоритма (блок-схемы)
Fig. 2. Representation of program code in the form of an algorithm (block-diagram)

Заключение

Приведенный выше вариант пропускной системы более удобен в эксплуатации по сравнению с вариантом использования одних ворот для пропуска всех автомобилей и посетителей, однако является более затратным, так как требует для себя больше пространства и, как следствие, больше материальных средств на его постройку, эксплуатацию и обслуживание [9].

Пропускная система с двумя шлагбаумами и отдельным входом для посетителей имеет определенную популярность из-за своей простоты в применении, обслуживании и монтаже. В силу своей распространенности – вопрос выбора для автоматизации именно для неё является наиболее целесообразным и является одним из первых решений в области технологий умного производства «Индустрия 4.0.» для контрольно – пропускных систем [10]. Такая система наглядно демонстрирует процесс пропуска автомобилей и людей на предприятиях, что делает её оптимальным примером для использования методики автоматного программирования, которая может быть употреблена для более технически сложных систем, имеющих узкие, специализированные области применения.

Список источников:

1. Канжелев С.Ю., Шалыто А.А. Преобразование графов переходов, представленных в формате MSVisio, в исходные коды программ для различных языков программирования (инструментальное средство MetaAuto). – Национальный исследовательский университет институт точной механики и оптики. – Санкт-Петербург: ИТМО, 2005. – 102 с.

2. Хайруллина Л.И., Хайруллин И.Р., Чижова М.А. Цифровизация в сфере производственной безопасности: основные аспекты вопроса // Век качества. – 2022. – С. 141-153.

References:

1. Kanzhelev S.Yu., Shalyto A.A. The Use of Graph Diagrams and Transition Graphs in MSVisio Format Into Program Source Codes for Various Programming Languages (MetaAuto Tool). Saint Petersburg: National Research University Institute of Fine Mechanics and Optics; 2005.

2. Khairullina L.I., Khairullin I.R., Chizhova M.A. Digitalization in the Field of Occupational Safety and Industrial Safety: The Main Aspects of the Issue. Age of Quality. 2022:141-153.

3. Шалыто А.А. Методы аппаратной и программной реализации алгоритмов. – Национальный исследовательский университет институт точной механики и оптики. – Санкт–Петербург: ИТМО, 2005. – 779 с.

4. Normand V., Exertier D., «Model-driven systems engineering: SysML & the MDSysE approach at Thales», in «Model Driven Engineering for distributed real-time embedded systems» // John Wiley & Sons. – 2005.

5. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Edition // Wiley. – 2015.

6. Татарчевский В.А. Switch – технология в задачах логического управления // Программные продукты и системы. – 2006. – №5. – С. 30-32.

7. Шалыто А.А., Туккель Н.И. Switch-технология – автоматный подход к созданию программного обеспечения «реактивных» систем // Программирование. – 2001. – №5. – С. 45-62.

8. Липкин Е. ИНДУСТРИЯ 4.0: Умные технологии – ключевой элемент в промышленной конкуренции. – М.: ООО «Остек–СМТ», 2017. – 224 с.

9. Шалыто А.А. Алгоритмизация и программирование задач логического управления. – СПбГУ ИТМО, 1998. – 55 с.

10.Поликарпова Н.И., Шалыто А.А. Автоматное программирование. Учебно-методическое пособие. – СПбГУ ИТМО, 2007. – 108 с.

3. Shalyto A.A. Methods for Hardware and Software Implementation of Algorithms. National Research University Institute of Precision Mechanics and Optics. Saint Petersburg: ITMO; 2005.

4. Normand V, Exertier D. Model-Driven Systems Engineering: SysML & the MDSysE Approach at Thales. In: Model Driven Engineering for Distributed Real-Time Embedded Systems. John Wiley & Sons; 2005.

5. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th ed. Wiley; 2015.

6. Tatarchevsky V.A. Switch – Technology in Logical Control Problems. Software Products and Systems. 2006;5:30-32.

7. Shalyto A.A., Tukkel N.I. Switch Technology – An Automated Approach to Developing Software for Reactive Systems. Programming. 2001;5:45-62.

8. Lipkin E. INDUSTRY 4.0: Smart Technologies are a Key Element in Industrial Competition. Moscow: Ostek-SMT; 2017.

9. Shalyto A.A. Algorithmization and Programming for Logic Control Tasks. Saint Petersburg State University ITMO; 1998.

10. Polikarpova N.I., Shalyto A.A. Automa-Based Programming. Saint Petersburg State University ITMO; 2007.

Информация об авторах:

Гумунюк Павел Васильевич
аспирант Института конструкторско-технологической информатики Российской академии наук

Шурпо Александр Николаевич
кандидат технических наук, старший научный сотрудник Института конструкторско-технологической информатики Российской академии наук, https://orcid.org/0000-0003-1962-1969

Information about the authors:

Gumunyuk Pavel Vasilievich
Graduate Student of the Institute for Design-Technological Informatics of the Russian Academy of Sciences

Shurpo Alexander Nikolaevich
Candidate of Technical Sciences, Senior Researcher at the Institute for Design-Technological Informatics of the Russian Academy of Sciences, https://orcid.org/0000-0003-1962-1969

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.
Contribution of the authors: the authors contributed equally to this article.

Авторы заявляют об отсутствии конфликта интересов.
The authors declare no conflicts of interests.

Статья поступила в редакцию 17.08.2023; одобрена после рецензирования 21.09.2023; принята к публикации 28.09.2023.

The article was submitted 17.08.2023; approved after reviewing 21.09.2023; accepted for publication 28.09.2023.

Рецензент – Медведев Д.М., кандидат технических наук, доцент, Брянский государственный технический университет.

Reviewer – Medvedev D.M., Candidate of Technical Sciences, Associate Professor, Bryansk State Technical University.