Н. Г. ТОПОЛЬСКИЙ, д-р техн. наук, профессор, заслуженный деятель науки РФ, профессор кафедры информационных технологий, Академия ГПС МЧС России (Россия, 129366, г. Москва, ул. Бориса Галушкина, 4; e-mail: ntopolskii@mail.ru)

А. В. КРЮЧКОВ, канд. техн. наук, доцент кафедры комплексной безопасности критически важных объектов, Российский государственный университет нефти и газа (национальный исследовательский университет) им. И. М. Губкина (Россия, 119991, г. Москва, Ленинский просп., д. 65, корп. 1; e-mail: kruchkov.a@gubkin.ru)

Д. С. ГРАЧЕВ, соискатель факультета подготовки научно-педагогических кадров, кафедра информационных технологий, Академия ГПС МЧС России (Россия, 129366, г. Москва, ул. Бориса Галушкина, 4)

К. А. МИХАЙЛОВ, адъюнкт факультета подготовки научно-педагогических кадров, кафедра информационных технологий, Академия ГПС МЧС России (Россия, 129366, г. Москва, ул. Бориса Галушкина, 4; e-mail: mihkir.94@mail.ru)

УДК 614.849:004.42

ОПРЕДЕЛЕНИЕ СОСТАВА ПРОГРАММ В СПЕЦИАЛЬНОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ ПОЖАРОВЗРЫВОБЕЗОПАСНОСТИ ОБЪЕКТОВ

Рассмотрены единичные программные системы (ЕПС) автоматизированных систем пожаровзрывобезопасности (АСПВБ) объектов. Дано одно из возможных математических определений программ ЕПС и их характеристик. Проведена оценка количества формул в специальном программном обеспечении (СПО) АСПВБ и автоматизированных систем управления технологическими процессами в целом. Описан процесс поиска унификатора в ЕПС на множестве формул СПО АСПВБ. Показано, что для его определения необходимо провести разбиение ЕПС на базовые группы функциональности, представляющие собой неделимые на части элементы.

Ключевые слова: автоматизированные системы управления технологическими процессами; специальное программное обеспечение; автоматизированные системы пожаровзрывобезопасности; единичная программная система; базовые группы функциональности.

DOI: 10.18322/PVB.2018.27.07-08.67-73

Введение

Научно-технический прогресс диктует все новые и новые требования к принципам работы производственных мощностей современных предприятий. Для их устойчивой работы в современных условиях используются автоматизированные системы управления технологическими процессами (АСУТП), составной частью которых является специальное программное обеспечение (СПО).

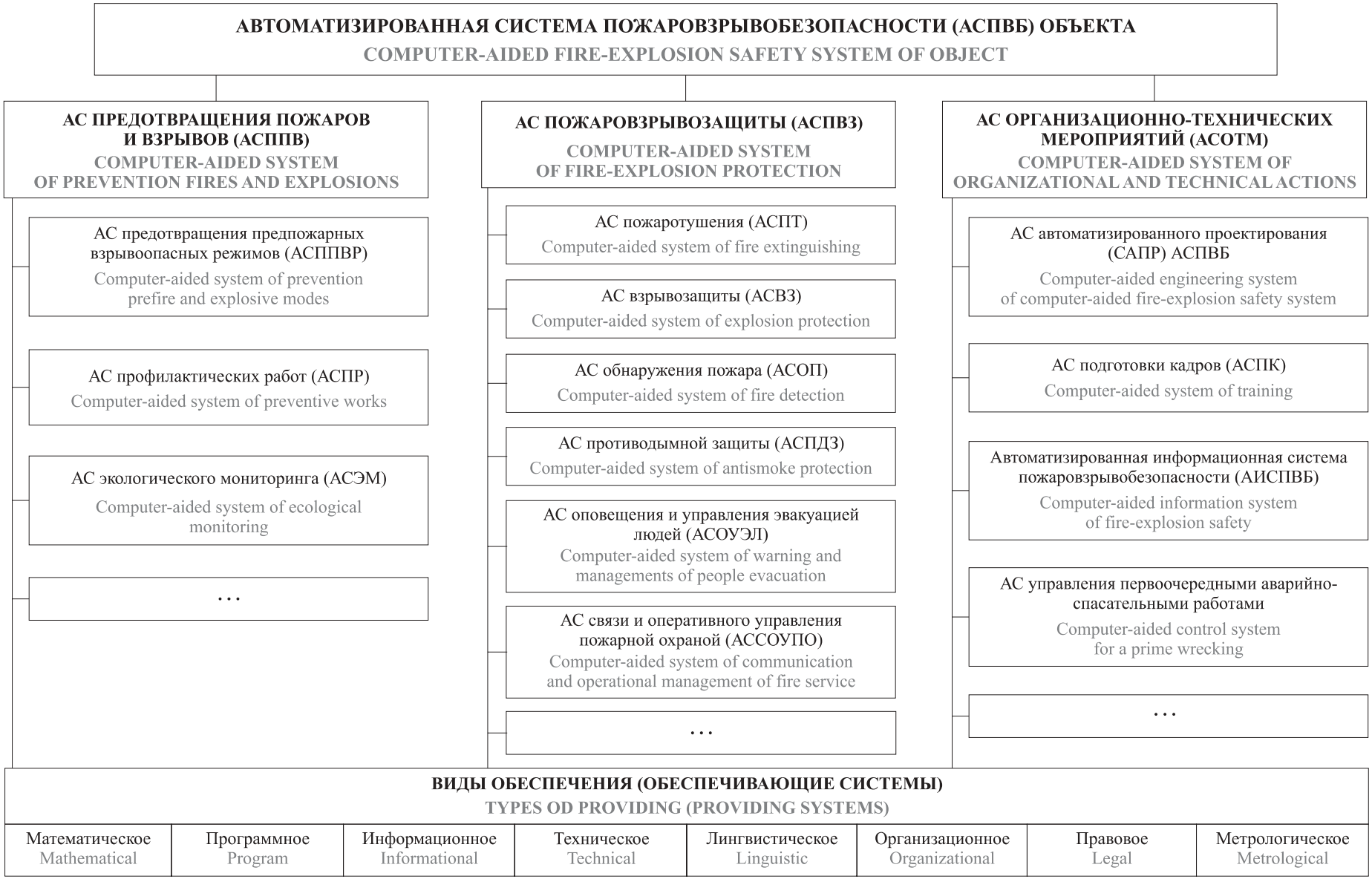
Таким образом, цель настоящей работы — определение состава программ в СПО автоматизированных систем пожаровзрывобезопасности объектов (АСПВБ). При этом поставлены и решены следующие задачи: описание математического определения единичных программных систем (ЕПС) и их характеристик; оценка количества формул в СПО автоматизированных систем пожаровзрывобезопасности; описание (с формальной точки зрения) поиска унификатора в единичной программной системе.

Общая постановка задачи исследования

Нефтеперерабатывающие производства (НПП) — важная составляющая современной промышленности. Они, помимо всего прочего, могут иметь в составе АСУТП автоматизированные системы пожаровзрывобезопасности [1, 2]. В данные системы входят обеспечивающие системы, а также функциональные автоматизированные системы (АС) нижестоящего уровня, включающие в себя [1, 2] (см. рисунок):

- АС предотвращения пожаров и взрывов (АСППВ);
- АС пожаровзрывозащиты (АСПВЗ);
- АС организационно-технических мероприятий (АСОТМ).

Как правило, устойчивая работа перечисленных АС обеспечивается в настоящее время за счет устойчивой работы СПО, поэтому особое внимание следует уделить методам его синтеза [3–14]. Учитывая, что СПО АСПВБ представляет собой сложный ин-



Общая схема автоматизированной системы пожаровзрывобезопасности объекта / General diagram of the computer-aided fire-explosion safety system of object

теллектуальный продукт, следует разделить его на составные части — единичные программные системы. ЕПС должны выполнять определенные функции операторов АСПВБ на автоматизированных рабочих местах (АРМ) в отдельных помещениях или цехах НПП. При этом влияние на ЕПС их разработчиков очень велико.

Общих функциональных задач у ЕПС нет, так как каждая из них отвечает за свой небольшой участок. Однако нужно, чтобы все они обеспечивали устойчивую работу АСПВБ в целом, особенно в определенных ситуациях. Предотвращение пожаров и взрывов на НПП — составная часть работы всех вместе взятых ЕПС в СПО АСПВБ. В связи с этим необходимо обеспечить такие условия синтеза ЕПС [4–6], которые позволят добиться нужной устойчивости в их работе. Для этого следует детально рассмотреть математическое определение программ ЕПС и их характеристик [15–20], по которым можно установить, насколько та или иная ЕПС устойчива к деструктивному влиянию людей, ее синтезирующих.

Теоретические основы

Поскольку ЕПС представляет собой программу, рассмотрим одно из ее возможных математических определений. Согласно [21] программа формально определена, если заданы входной вектор (перечень входных данных), выходной вектор (перечень выходных данных), внутрипрограммный вектор (перечень переменных, используемых для промежуточных вычислений) и ориентированный граф, который определяет порядок перехода от входных данных к выходным через промежуточные:

$$\{IN_p,\ OUT_p,\ SOLE_p,\ (U_p,\ R_p)\}, \tag{1}$$

где IN_p — входной вектор;

OUT_p — выходной вектор;

$SOLE_p$ — внутрипрограммный вектор;

(U_p, R_p) — ориентированный граф;

U_p — непустое множество узлов (вершин) графа состояний;

R_p — непустое множество ребер (стрелок), нагруженных данными о вероятности развития процесса по данному ребру.

При этом должно существовать отображение направленной инцидентности $D(r)$, определенное на множестве пар узлов (вершин) графа R_p, суть которого состоит в определении возможных переходов между вершинами:

$$\exists\, D(r)\,(U_p,\ U_p), \tag{2}$$

где

$$(U_p,\ U_p) \in R_p. \tag{3}$$

Семантика программной конструкции может быть также описана с помощью двух предикатов [19, 20]:

$$\begin{gathered} W_p\,(S_p,\ R_p) \text{ (предусловие);} \\ R_p \text{ (постусловие),} \end{gathered} \tag{4}$$

где $W_p\,(S_p,\ R_p)$ — множество начальных состояний (предусловие);

R_p — множество состояний, возникающих в результате применения оператора (предиката) S_p в качестве преобразователя множества начальных состояний (постусловие).

Если говорить о какой-то одной предметной области, то представленные выше формулы будут моделировать одну из ее функциональных характеристик. Для математического описания спецификации (функциональной характеристики) может рассматриваться следующая формула [21]:

$$F\,(a) \underset{P\,(a)}{<=} Z \ \text{ при } R(a,\ Z), \tag{5}$$

где $F(a)$ — спецификация одного из требований, описывающих единичную характеристику предметной области;

a — вход программной конструкции;

Z — выход программной конструкции;

$P(a)$ — входное условие;

$R(a,\ Z)$ — выходное условие.

В свою очередь программная конструкция, выводимая из такой спецификации, может быть представлена как предикат вида

$$(\forall a)\,(\exists Z)\ \text{ если }\ P(a)\ \text{ то }\ R(a,\ Z). \tag{6}$$

С формальной точки зрения “унификация — это процесс отыскания общей части двух выражений” [21]. В [21] приводится формальное определение спецификации алгоритма унификации. Алгоритм унификации — это процедура отыскания наиболее общего унификатора для двух выражений, если для них вообще таковой существует. Унификатор — это набор сходных характеристик группы выражений [21].

Такой подход вполне оправдан, когда речь идет о небольших программных системах (не более 100 строк). При эксплуатации СПО АСПВБ и АСУТП НПП в целом для заданного класса задач приходится иметь дело с бóльшим (как правило, значительно более 100) числом АРМ, каждое из которых представляет собой набор программ с числом строк до 30 тысяч. В связи с этим алгоритм унификации применим лишь при рассмотрении небольшой в смысле количества функциональной характеристики ЕПС, а для СПО АСПВБ и АСУТП НПП в целом его следует применять с большой осторожностью.

Оценим количество формул в СПО. Примем в качестве допущения, что объем кода в ЕПС немного свыше 10 тысяч строк. Исходя из расчетов, выполненных авторами, можно утверждать, что каждая реализация СПО содержит набор из 100 и более программ, семантика программной конструкции кото-

рых определяется формулой (4), а самих программ — формулой (1). При проведении суперпозиции этих моделей возможно получение набора из 100 формул на одну программу. Для СПО АСПВБ и АСУТП в целом необходимо около миллиона формул. Имея матрицу такого размера, будет сложно провести расчеты, но алгоритм унификации всей группы ЕПС в СПО АСПВБ (поиск унификатора данной группы) следует рассматривать как правильное направление по повышению его устойчивости.

Таким образом, в терминах определения программ, приводимых в [21], предварительная задача исследования может быть определена как поиск унификатора на множестве формул СПО АСПВБ. В ней могут рассматриваться две подзадачи: поиск унификаторов каждой ЕПС на множестве функциональных характеристик предметной области и поиск унификатора СПО на множестве всех найденных унификаторов ЕПС.

Для определения унификатора в ЕПС необходимо провести разбиение ЕПС на базовые группы функциональности (БГФ), каждая из которых будет представлять собой неразбиваемую на части программу, удовлетворяющую условиям приведенных выше формул. Если рассматривать ЕПС с таких позиций, то ее можно представить множеством

$$\{F_1,\ F_2,\ \ldots,\ F_M\}, \tag{7}$$

где $F_1,\ F_2,\ \ldots,\ F_M$ — набор базовых групп функциональности ЕПС.

Тогда, учитывая, что M — число групп функциональности ЕПС, а N — число ЕПС в АСПВБ, СПО можно представить как суперпозицию всех ЕПС. В этом случае мы получаем матрицу “неочищенных” базовых групп функциональности, для которых необходимо провести унификацию:

$$\begin{gathered} F_{11},\ F_{21},\ \ldots,\ F_{M1};\\ F_{12},\ F_{22},\ \ldots,\ F_{M2};\\ \ldots\\ \ldots\\ F_{1N},\ F_{2N},\ \ldots,\ F_{MN}. \end{gathered} \tag{8}$$

В этом случае с формальной точки зрения для поиска унификатора группы ЕПС необходимо выделить только такие базовые группы функциональности различных ЕПС, которые будут обладать сходными характеристиками. Таким образом, выделенные в (8) группы функциональности будут отнесены к базовым. Другими словами, это могут быть базовые элементы программ, соответствующие базовым группам функциональности. Их набор позволит сформировать пространство групп функциональности ЕПС. В этом случае мы получим аналогичную (8) матрицу базовых элементов программ, которым могут соответствовать функции библиотек подпрограмм или элементы репозитория компонентов:

$$\begin{gathered} F_{1\text{Б}},\ F_{2\text{Б}},\ \ldots,\ F_{M\text{Б}1};\\ F_{1\text{Б}},\ F_{2\text{Б}},\ \ldots,\ F_{M\text{Б}2};\\ \ldots\\ \ldots\\ F_{1\text{Б}},\ F_{2\text{Б}},\ \ldots,\ F_{M\text{Б}N}. \end{gathered} \tag{9}$$

Часть групп функциональности (функций) разных ЕПС может быть реализована одинаково с помощью базовых элементов программ, а часть не может, так как каждая из ЕПС используется для определенной предметной области. Множество таких базовых групп функциональности представляет собой унификатор группы ЕПС, который можно определить как

$$\{F_{1\text{Б}},\ F_{2\text{Б}},\ \ldots,\ F_{K\text{Б}}\} \in R_f, \tag{10}$$

где $F_{1\text{Б}}, F_{2\text{Б}}, \ldots, F_{K\text{Б}}$ — набор из K базовых групп функциональности;

R_f — пространство групп функциональности.

Отсюда следует, что выявление групп базовых элементов программ позволяет однозначно снизить влияние человека на СПО АСПВБ и, следовательно, повысить его надежность [22, 23].

Заключение

Таким образом, задача определения состава программ в СПО АСПВБ объектов может быть определена как поиск подмножества базовых элементов программ (БЭП) на пространстве групп функциональности единичных программных систем. Ее решением будет множество групп БЭП на дискретном пространстве ЕПС. Суперпозиция групп БЭП и некоторых дополнительных унифицированных программных элементов (конфигураций, компонент, процедур и функций, элементов интерфейса, баз данных и т. п.), специфицирующих данное обобщение для конкретной предметной области, создаст объект ЕПС. Недостатками такой математической конструкции являются большая размерность матрицы (9) и отсутствие учета иерархий наборов требований. В связи с этим логичным продолжением процесса математической постановки задачи определения состава программ в СПО автоматизированных систем пожаровзрывобезопасности объектов может стать моделирование иерархических характеристик СПО с помощью графов.

СПИСОК ЛИТЕРАТУРЫ

1. *Топольский Н. Г.* Основы автоматизированных систем пожаровзрывобезопасности объектов. — М. : МИПБ МВД России, 1997. — 164 с.
2. *Абросимов А. А.*, *Топольский Н. Г.*, *Фёдоров А. В.* Автоматизированные системы пожаровзрывобезопасности нефтеперерабатывающих производств. — М. : МИПБ МВД России, 1999. — 239 с.
3. *Гаплаев А. А.-Б.* Автоматизированный комплекс контроля и испытаний системы управления противопожарной защиты нефтеперерабатывающих производств : дис. … канд. техн. наук. — М., 2018. — 226 с.
4. *Крючков А. В.* Обобщение опыта синтеза специального программного обеспечения на различных инструментальных средствах // Технологии техносферной безопасности. — 2015. — Вып. 3(61). — С. 252–263. URL: http://agps-2006.narod.ru/ttb/2015-3/41-03-15.ttb.pdf (дата обращения: 03.06.2018).
5. *Gajski D. D.*, *Abdi S.*, *Gerstlauer A.*, *Schimer G.* Software synthesis // Embedded System Design. — Boston, MA : Springer, 2009. — P. 155–197. DOI: 10.1007/978-1-4419-0504-8_5.
6. *Крючков А. В.* Методология универсализации синтеза специального программного обеспечения крупной автоматизированной системы управления предприятием // Технологии техносферной безопасности. — 2015. — Вып. 3(61). — С. 264–268. URL: http://agps-2006.narod.ru/ttb/2015-3/40-03-15.ttb.pdf (дата обращения: 03.06.2018).
7. *Stolz V.* Special issue: Harnessing theories for tool support in software // Innovations in Systems and Software Engineering. — 2013. — Vol. 9, Issue 1. — P. 1–2. DOI: 10.1007/s11334-012-0193-4.
8. *Макконнелл С.* Профессиональная разработка программного обеспечения: сокращение сроков, повышение качества продукта, больше удачных проектов, расширение возможностей успешной карьеры / Пер. с англ. — СПб. : Символ&Плюс, 2006. — 240 с.
9. *Вендров А. М.* Проектирование программного обеспечения экономических информационных систем. — М. : Финансы и статистика, 2006. — 545 с.
10. *Zheng Qin*, *Xiang Zheng*, *Jiankuan Xing*. Software Architecture // Advanced Topics in Science and Technology in China. — Berlin, Heidelberg : Springer, 2008. — 337 p. DOI: 10.1007/978-3-540-74343-9.
11. *Bass L.*, *Clements P.*, *Kazman R.* Software architecture in practice. — 3rd Edition. — Addison-Wesley Professional, 2012. — 640 p.
12. *Rozanski N.*, *Woods E.* Software systems architecture: working with stakeholders using viewpoints and perspectives. — 2nd Edition. — Addison-Wesley Professional, 2012. — 704 p.
13. *Almeida J. B.*, *Frade M. J.*, *Pinto J. S.*, *Melo de Sousa S.* Rigorous software development. — London : Springer-Verlag, 2011. — 307 p. DOI: 10.1007/978-0-85729-018-2.
14. *Crookshanks E.* Practical Software Development Techniques: Tools and techniques for building enterprise software. — Berkeley, CA : Apress, 2014. — 212 p. DOI: 10.1007/978-1-4842-0728-4.
15. *Грис Д.* Наука программирования / Пер. с англ.; под ред. А. П. Ершова. — М. : Мир, 1984. — 416 с.
16. *Чень Ч.*, *Ли Р.* Математическая логика и автоматическое доказательство теорем / Пер. с англ.; под ред. С. Ю. Маслова. — М. : Наука, 1983. — 360 с.
17. *Дийкстра Э., Больe Л., Хоор К., Дал У.-И.* Языки программирования / Ред. Ф. Женюи; пер. с англ. В. П. Кузнецова, под ред. В. М. Курочкина. — М. : Мир, 1972. — 410 с.
18. *Умрихин Ю. Д.* Оптимизация сложных информационных систем. — М. : Министерство радиопромышленности СССР, 1983. — 125 с.
19. *Клименко И. С.* Теория систем и системный анализ. — М. : РосНОУ, 2014. — 264 с.
20. *Шиханович Ю. А.* Введение в современную математику. Начальные понятия. — М. : Наука, 1965. — 376 с.
21. *Ильин В. Д.* Система порождения программ. — М. : Наука, 1989. — 257 с.
22. *De Lucia A.*, *Ferrucci F.* Software engineering. — Berlin, Heidelberg : Springer-Verlag, 2013. — 237 p. DOI: 10.1007/978-3-642-36054-1.
23. *Peled D. A.* Software reliability methods. — New York : Springer-Verlag, 2001. — 332 p. DOI: 10.1007/978-1-4757-3540-6.

Материал поступил в редакцию 15 июня 2018 г.

Для цитирования: *Топольский Н. Г., Крючков А. В., Грачев Д. С., Михайлов К. А.* Определение состава программ в специальном программном обеспечении автоматизированных систем пожаровзрывобезопасности объектов // Пожаровзрывобезопасность / Fire and Explosion Safety. — 2018. — Т. 27, № 7–8. — С. 67–73. DOI: 10.18322/PVB.2018.27.07-08.67-73.

DETERMINATION OF THE COMPOSITION OF PROGRAMS IN A SPECIAL SOFTWARE OF THE COMPUTER-AIDED FIRE-EXPLOSION SAFETY SYSTEMS OF OBJECTS

TOPOLSKIY N. G., Doctor of Technical Sciences, Professor, Honoured Science Worker of Russian Federation, Professor of Department of Information Technology, State Fire Academy of Emercom of Russia (Borisa Galushkina St., 4, Moscow, 129366, Russian Federation; e-mail: ntopolskii@mail.ru)

KRYUCHKOV A. V., Candidate of Technical Sciences, Assistant Professor of Department of Complex Security of Critical Objects, Gubkin Russian State University of Oil and Gas (National Research University) (Leninskiy Avenue, 65, Bldg. 1, Moscow, 119991, Russian Federation; e-mail: kruchkov.a@gubkin.ru)

GRACHEV D. S., Competitor of Faculty of Scientific and Pedagogical Staff, Department of Information Technology, State Fire Academy of Emercom of Russia (Borisa Galushkina St., 4, Moscow, 129366, Russian Federation)

MIKHAYLOV K. A., Postgraduate Student of Faculty of Scientific and Pedagogical Staff, Department of Information Technology, State Fire Academy of Emercom of Russia (Borisa Galushkina St., 4, Moscow, 129366, Russian Federation; e-mail: mihkir.94@mail.ru)

ABSTRACT

Introduction. For the sustainable operation of modern industries, in particular oil refineries, computer-aided process control systems are used. One of the key components of these systems are the computer-aided fire-explosion safety systems. A crucial role in these systems is played by special software, the reliability of which depends on the successful functioning of the computer-aided fire-explosion safety systems and in general oil refineries.

The work purpose is to determine the composition of programs in special software for computer-aided fire-explosion safety systems of objects. The following tasks are solved: the mathematical definition of single software systems and their characteristics is described; the number of formulas in the special software of computer-aided fire-explosion safety systems is estimated; from the formal point of view, the identification of the unifier in a single software system is described.

Methods. The article uses the methods of mathematical logic, graph theory for the formal description of single software systems and their functional characteristics.

Results and discussion. One of the possible mathematical definitions of single software systems of computer-aided fire-explosion safety systems and their characteristics is given. The assesment of quantity of formulas for the special software of the computer-aided fire-explosion safety systems and computer-aided process control systems in general which about one million is necessary is carried out. It is shown that to determine the unifier in a single program system, it is necessary to divide it into basic groups of functionality. The basic functionality groups are meant as basic elements of programs.

Conclusions. In the end, the identification of groups of basic elements of programs allows to define unambiguously the required structure of programs of the special software of the computer-aided fire-explosion safety systems of objects, increase its reliability and simplify the work with it.

Keywords: computer-aided process control systems; computer-aided fire-explosion safety systems; synthesis of special software; single software system; basic functionality groups.

REFERENCES

1. Topolskiy N. G. *Osnovy avtomatizirovannykh sistem pozharovzryvobezopasnosti obyektov* [Basics of computer-aided fire and explosion safety systems]. Moscow, Fire Safety Institute of the Ministry of Internal Affairs of Russia Publ., 1997. 164 p. (in Russian).
2. Abrosimov A. A., Topolskiy N. G., Fedorov A. V. *Avtomatizirovannyye sistemy pozharovzryvobezopasnosti neftepererabatyvayushchikh proizvodstv* [Computer-aided fire and explosion safety systems of petroleum refineries]. Moscow, State Fire Academy of the Ministry of Internal Affairs of Russia Publ., 1999. 239 p. (in Russian).
3. Gaplaev A. A.-B. *Computer-aided complex of control and testing of management systems of oil refineries fire protection*. Cand. tech. sci. diss. Moscow, 2018. 226 p. (in Russian).

4. Kruchkov A. V. Summarizing the experience of synthesis special software for different programming languages. *Tekhnologii tekhnosfernoy bezopasnosti / Technology of Technosphere Safety*, 2015, issue 3(61), pp. 252–263 (in Russian). Available at: http://agps-2006.narod.ru/ttb/2015-3/41-03-15.ttb.pdf (Accessed 3 June 2018).
5. Gajski D. D., Abdi S., Gerstlauer A., Schimer G. Software synthesis. In: *Embedded System Design*. Boston, MA, Springer, 2009, pp. 155–197. DOI: 10.1007/978-1-4419-0504-8_5.
6. Kruchkov A. V. Universal application synthesis methodology of special software for large automated enterprise control system. *Tekhnologii tekhnosfernoy bezopasnosti / Technology of Technosphere Safety*, 2015, issue 3(61), pp. 264–268 (in Russian). Available at: http://agps-2006.narod.ru/ttb/2015-3/40-03-15.ttb.pdf (Accessed 3 June 2018).
7. Stolz V. Special issue: Harnessing theories for tool support in software. *Innovations in Systems and Software Engineering*, 2013, vol. 9, issue 1, pp. 1–2. DOI: 10.1007/s11334-012-0193-4.
8. McConnell S. *Professional software development: shorter schedules*, *higher quality products*, *more successful projects*, *enhanced careers*. Addison-Wesley, 2004. 243 p. (Russ. ed.: McConnell S. Professionalnaya razrabotka programmnogo obespecheniya: sokrashcheniye srokov, povysheniye kachestva produkta, bolshe udachnykh proyektov, rasshireniye vozmozhnostey uspeshnoy karyery. Saint Petersburg, Simvol&Plyus Publ., 2006. 240 p.).
9. Vendrov A. M. *Proyektirovaniye programmnogo obespecheniya ekonomicheskikh informatsionnykh sistem* [The software design of economic information systems]. Moscow, Finansy i statistika Publ., 2006. 545 p. (in Russian).
10. Zheng Qin, Xiang Zheng, Jiankuan Xing. Software Architecture. In: *Advanced Topics in Science and Technology in China*. Berlin, Heidelberg, Springer, 2008. 337 p. DOI: 10.1007/978-3-540-74343-9.
11. Bass L., Clements P., Kazman R. *Software architecture in practice*. 3rd Edition. Addison-Wesley Professional, 2012. 640 p.
12. Rozanski N., Woods E. *Software systems architecture: working with stakeholders using viewpoints and perspectives*. 2nd Edition. Addison-Wesley Professional, 2012. 704 p.
13. Almeida J. B., Frade M. J., Pinto J. S., Melo de Sousa S. *Rigorous software development*. London, Springer-Verlag, 2011. 307 p. DOI: 10.1007/978-0-85729-018-2.
14. Crookshanks E. *Practical Software Development Techniques: Tools and techniques for building enterprise software*. Berkeley, CA, Apress, 2014. 212 p. DOI: 10.1007/978-1-4842-0728-4.
15. Gries D. *The science of programming*. New York, Heidelberg, Berlin, Springer-Verlag, 1981 (Russ. ed.: Gries D. Nauka programmirovaniya. Moscow, Mir Publ., 1984. 416 p.).
16. Chang Ch.-L., Lee R. *Symbolic logic and mechanical theorem proving*. New York, San Francisco, London, Academic Press, 1973 (Russ. ed.: Chang Ch.-L., Lee R. Matematicheskaya logika i avtomaticheskoye dokazatelstvo teorem. Moscow, Nauka Publ., 1983. 360 p.)
17. Dijkstra E. W., Beaulieu L., Hoare C. A. R., Dahl O.-J. *Programming languages*. London, 1968 (Russ. ed.: Dijkstra E. W., Beaulieu L., Hoare C. A. R., Dahl O.-J. Yazyki programmirovaniya. Moscow, Mir Publ., 1972. 410 p.).
18. Umrikhin Yu. D. *Optimizatsiya slozhnykh informatsionnykh system* [Optimization of complex information systems]. Moscow, Ministry of Radio Technology Publ., 1983. 125 p. (in Russian).
19. Klimenko I. S. *Teoriya sistem i sistemnyy analiz* [Systems theory and system analysis]. Moscow, RosNOU Publ., 2014. 264 p. (in Russian).
20. Shikhanovich Yu. A. *Vvedeniye v sovremennuyu matematiku. Nachalnyye ponyatiya* [Introduction to modern mathematics: Initial concepts]. Moscow, Nauka Publ., 1965. 376 p. (in Russian).
21. Ilin V. D. *Sistema porozhdeniya programm* [The system of program generating]. Moscow, Nauka Publ., 1989. 257 p. (in Russian).
22. De Lucia A., Ferrucci F. *Software engineering*. Berlin, Heidelberg, Springer-Verlag, 2013. 237 p. DOI: 10.1007/978-3-642-36054-1.
23. Peled D. A. *Software reliability methods*. New York, Springer-Verlag, 2001. 332 p. DOI: 10.1007/978-1-4757-3540-6.

For citation: Topolskiy N. G., Kryuchkov A. V., Grachev D. S., Mikhaylov K. A. Determination of the composition of programs in a special software of the computer-aided fire-explosion safety systems of objects. *Pozharovzryvobezopasnost / Fire and Explosion Safety*, 2018, vol. 27, no. 7–8, pp. 67–73 (in Russian). DOI: 10.18322/PVB.2018.27.07-08.67-73.