

УДК 004.056

А. А. Корниенко, С. В. Диасамидзе**ПОДХОД К СТРУКТУРИРОВАНИЮ МЕТРИК СЛОЖНОСТИ
ДЛЯ ВЫЯВЛЕНИЯ ДЕФЕКТОВ БЕЗОПАСНОСТИ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Дата поступления: 19.05.2016

Решение о публикации: 17.08.2016

Цель: Обосновать применение метрик сложности программ в качестве дополнительного инструмента для выявления дефектов безопасности при проведении статического анализа программ. **Методы:** Применены методы программометрии, на основе которых разработан метод выявления дефектов безопасности в программных средствах с использованием модели программы в виде схемы Янова, а также совокупности структурированных метрик сложности. **Результаты:** Рассмотрены два взаимодополняющих направления формирования информативного структурированного метрического базиса. Первый заключается в ранжировании метрик сложности на основе экспертного оценивания в задачах выявления дефектов безопасности программного обеспечения. Во втором предлагается опираться на результаты экспериментального исследования чувствительности компонент расширенного метрического базиса к потенциально опасным программным конструкциям. **Практическая значимость:** Результат структурирования метрик сложности позволяет сформировать достаточно информативный метрический базис, состоящий из небольшого количества используемых метрик. Аппарат метрического контроля можно использовать при испытаниях программных средств по требованиям безопасности информации.

Информационная безопасность, программное обеспечение, метрика сложности, дефект безопасности, недеklarированные возможности, статический анализ.

Anatoliy A. Korniyenko, Dr. Sci. (Eng.), professor, department chair, kaa.pgups@ya.ru; ***Svetlana V. Diasamidze**, Cand. Sci. (Eng.), associate professor, sv.diass99@ya.ru (Petersburg State Transport University) APPROACH TO STRUCTURING COMPLEXITY METRICS FOR DETECTION OF SOFTWARE SECURITY FLAWS

Objective: To justify application of program complexity metrics as an additional instrument for detecting security flaws when conducting static code analysis. **Methods:** Program metrics methods were applied, and on their basis a method was developed for detecting security flaws in software with the use of program model presented as Yanov schemata, as well as of sum total of structured complexity metrics. **Results:** Two complementary directions for forming informative structured metric base were considered. The first one constitutes ranging complexity metrics on the basis of expert evaluation with the task of determining software security flaws. The second one proposes to rely on results of experimental study of sensitivity of components of expanded metric base for potentially dangerous programming constructs. **Practical importance:** Results of structuring of complexity metrics allows for forming sufficiently informative metrical base consisting of a small number of metrics applied. Metric control apparatus may be applied in testing software on information security requirements.

Information security, software, complexity metrics, security flaw, undocumented features, static analysis.

В современных условиях для поддержания высокого уровня безопасности движения, грузовых и пассажирских перевозок и других критичных технологических процессов и систем важное значение приобретают вопросы обеспечения функциональной и информационной безопасности информационно-управляющих и автоматизированных систем железнодорожного транспорта и программного обеспечения (ПО) как их неотъемлемой составляющей. Для оценки соответствия по требованиям функциональной безопасности ПО построена четырехуровневая система показателей качества безопасности, третий уровень которой представлен метриками качества и сложности программ [8].

В настоящее время в мировой практике используются несколько сотен показателей, в той или иной степени описывающих сложность программ. Однако при реальном использовании метрик исследователи часто пытались максимизировать их универсальность без учета области применения исследуемых программных средств (ПС), что существенно снизило значимость метрик сложности программ как эффективного инструмента оценивания свойств программы на разных этапах жизненного цикла.

Тем не менее, класс метрик для оценивания топологической и информационной сложности программ может с успехом применяться в качестве дополнительного инструмента для выявления дефектов безопасности (ДБ) при статическом анализе программ [3, 6].

Выявление дефектов безопасности ПО (потенциально опасных конструкций (ПОК) и уязвимостей программного кода, недеklarированных возможностей, программных закладок и т. п.) на этапе статического анализа связывается с использованием структурированных метрик сложности, с формированием метрического базиса, чувствительного к преднамеренным и непреднамеренным программным искажениям и к выбору критериев обнаружения и распознавания ДБ [2, 4].

Отметим, что структурирование метрик сложности, формирование метрического ба-

зиса и решающих правил важно и для задачи динамического контроля целостности [11].

При структурировании метрик сложности и формировании расширенного метрического базиса для оценивания безопасности программ на первом этапе необходимо связать модель программы, выбранную на основе анализа и упорядочения семантик, и совокупность метрик, характеризующих вычислительную структуру контролируемых программ и свойства вычислимости. Такое сопоставление на данном этапе проведено из эвристических соображений, исходя из смыслового содержания метрик топологической и информационной сложности и предположений об их чувствительности к дефектам программного кода, в основе которых лежат топологические характеристики граф-модели программы, а также структуры и модели данных.

Сформированный таким образом расширенный метрический базис может включать до двух десятков достаточно информативных (с точки зрения обнаружения ДБ) метрик топологической и информационной сложности. Дальнейшее структурирование метрик сложности для формирования информативного, но с небольшим количеством наиболее чувствительных к искажениям программ компонент метрического базиса может идти по двум направлениям, дополняющим друг друга:

- 1) ранжирование метрик сложности на основе экспертного оценивания их чувствительности к ДБ;
- 2) экспериментальное исследование чувствительности совокупности структурированных метрик сложности к ПОК, имеющимся или внедряемым в программу.

Эти направления формирования информативного структурированного метрического базиса рассмотрены в данной работе.

Формирование расширенного метрического базиса для выявления дефектов безопасности программ

На основе исследований [2, 3, 7, 11] для анализа качества и безопасности программ

предложен метрический базис $M = \{M_1, M_2\}$, где M_1, M_2 – множества показателей, описывающих, соответственно, метрические базы топологической и информационной сложности. Анализ всех топологических метрик выявил набор метрик, которые могут быть спроецированы на структурированный базис конструкций языка ассемблер и позволяют создать метрический базис структуры программы $M_1 = \{m_{11}, m_{12}, \dots, m_{19}\}$ [10, 17]. Совокупность метрик $M_2 = \{m_{21}, m_{22}, \dots, m_{25}\}$ от-

ражает информационные свойства программ и составляет метрический базис информационной сложности программ [14] (табл. 1).

Структурирование предложенного метрического базиса на основе экспертного оценивания

В зависимости от целей и углубленности проводимого исследования программы мож-

ТАБЛИЦА 1. Расширенный метрический базис

Показатель	Метрика
Метрический базис топологической сложности программ	
m_{11}	Метрика Джилба $f^* = N_{SV}^* / (L_{loop} + L_{if} + L_{sub})$
m_{12}	Метрика Чена $M(G) = (n(G), N, Q_0)$
m_{13}	Метрика Вудворда (узловая мера) $Y(X)$
m_{14}	Метрика Пивоварского $N(G) = V^*(G) + \sum_i P_i$
m_{15}	Метрика точек пересечения $\min(a, b) < \min(p, q) < \max(a, b) \& \max(p, q) > \max(a, b) \mid \min(a, b) < \max(p, q) < \max(a, b) \& \min(p, q) < \min(a, b)$
m_{16}	Функциональная метрика Харрисона – Мейджела $f^* = Nc_1 / f_1$, где $f_1 = \sum c_1$
m_{17}	Метрика регулярных выражений $P(G) = N + L + \sum k$
m_{18}	Метрика Пратта $M(\{F_1, F_2, \dots, F_n\}) = \sum_{i=1}^n M(F_i)$ $M(IF_P_THEN_F_1_ELSE_F_2) = 2_MAX(M(F_1), M(F_2));$ $M(WHILE_P_DO_F) = 2M(F)$
m_{19}	Метрика Мак – Клуря $M(P) = fp^* X(P) + gp^* Y(P)$
Метрический базис информационной сложности программ	
m_{21}	Метрики размера программы $n = n_1 + n_2; N = N_1 + N_2; N^* = n_1 \log_2 n_1 + n_2 \log_2 n_2$
m_{22}	Информационная энтропия $H(x) = -\sum_{i=1}^n p(i) \log_2 p(i)$
m_{23}	Интеллектуальное содержание $I = V \times L$
m_{24}	Метрика Чепина $Q = P + 2M + 3C + 0,5T$
m_{25}	Метрика Спена $SP_{cp} = \frac{1}{n} \sum_{i=1}^n SP_i$

но применять определенную совокупность метрик сложности, составленную из элементов предложенного базиса с последующим ранжированием. При этом наряду с традиционными методами оценивания метрических показателей качества ПС используются методы экспертного оценивания. Наиболее распространены методы анализа иерархий Т. Саати [12] и принятия решений и обработки нечисловой информации по Н. В. Хованову [13]. Тем не менее, с учетом сложности в представлении неопределенности в суждениях экспертов и большого числа матриц попарных сравнений, для низкоуровневой оценки и ранжирования наиболее оправдан подход на основе математической теории свидетельств (теории Демпстера – Шейфера) [15, 18].

Критериями экспертного оценивания при ранжировании метрик сложности, составляющих метрический базис, выступали правила комбинирования теории свидетельств. Основными элементами служат начальные количественные весовые показатели метрик сложности – основные массы вероятности, определяющие субъективную степень уверенности; функция уверенности как общая степень уверенности и ее фокальные элементы с их базовыми вероятностями, полученными в данном случае на основе свидетельств по правилу Демпстера [1, 5].

Это правило позволяет для каждой совокупности фокальных элементов на всем множестве исходных данных сформировать результирующие подмножества и вычислить для них степени уверенности (результирующие комбинированные массы вероятностей), т. е. получить обобщенные весовые показатели метрик сложности, составляющих метрический базис для оценивания безопасности программы.

Для оценивания и ранжирования метрик сложности были привлечены три эксперта ($\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$), которые определяли весовые показатели для метрик, составляющие метрический базис (табл. 1). Таким образом, рассматривалось множество из 14 альтернатив $A = \{a_1, a_2, \dots, a_9, a_{10}, \dots, a_{14}\}$, соответствующее метрическому базису $M = \{m_{11}, m_{12}, \dots, m_{19}$,

$m_{21}, \dots, m_{25}\}$, которым эксперты присвоили следующие количественные весовые показатели (основные массы вероятности):

$$\mathcal{E}_1: p_1\{a_1\} = 0,15; p_1\{a_2, a_3, a_4, a_6\} = 0,1; p_1\{a_5\} = 0,15; p_1\{a_7, a_8\} = 0,08; p_1\{a_9\} = 0,06; p_1\{a_{10}\} = 0,18; p_1\{a_{12}\} = 0,2; p_1\{a_{11}, a_{13}, a_{14}\} = 0,08.$$

$$\mathcal{E}_2: p_2\{a_1\} = 0,18; p_2\{a_2, a_3, a_4, a_6\} = 0,06; p_2\{a_5\} = 0,19; p_2\{a_7, a_8\} = 0,02; p_2\{a_9\} = 0,03; p_2\{a_{10}\} = 0,3; p_2\{a_{12}\} = 0,2; p_2\{a_{11}, a_{13}, a_{14}\} = 0,02.$$

$$\mathcal{E}_3: p_3\{a_1\} = 0,2; p_3\{a_2, a_3, a_4, a_6\} = 0,08; p_3\{a_5\} = 0,2; p_3\{a_7, a_8\} = 0,06; p_3\{a_9\} = 0,08; p_3\{a_{10}\} = 0,15; p_3\{a_{12}\} = 0,15; p_3\{a_{11}, a_{13}, a_{14}\} = 0,08.$$

Уже на этом этапе, исходя из веса присвоенных экспертами начальных масс вероятностей, можно предварительно объединить элементы в группы со следующими весовыми показателями:

$$\mathcal{E}_1: p_1\{a_1, a_5, a_{10}, a_{12}\} = 0,68; p_1\{a_2, a_3, a_4, a_6, a_9\} = 0,16; p_1\{a_7, a_8, a_{11}, a_{13}, a_{14}\} = 0,16;$$

$$\mathcal{E}_2: p_2\{a_1, a_5, a_{10}, a_{12}\} = 0,87; p_2\{a_2, a_3, a_4, a_6, a_9\} = 0,09; p_2\{a_7, a_8, a_{11}, a_{13}, a_{14}\} = 0,04;$$

$$\mathcal{E}_3: p_3\{a_1, a_5, a_{10}, a_{12}\} = 0,7; p_3\{a_2, a_3, a_4, a_6, a_9\} = 0,16; p_3\{a_7, a_8, a_{11}, a_{13}, a_{14}\} = 0,14.$$

Далее для получения обобщенной оценки были скомбинированы начальные значения вероятностей для всех пар экспертов, определены непустые пересечения маргинальных фокальных элементов, рассчитаны степень конфликтности и константа нормализации, нормализованы комбинированные массы вероятностей.

Полученные результирующие комбинированные основные назначения вероятностей экспертов $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ имеют следующие значения:

$$\{m_{11}, m_{15}, m_{21}, m_{23}\} - p_{123}\{a_1, a_5, a_{10}, a_{12}\} = 0,7441;$$

$$\{m_{12}, m_{13}, m_{14}, m_{16}, m_{19}\} - p_{123}\{a_2, a_3, a_4, a_6, a_9\} = 0,1636;$$

$$\{m_{17}, m_{18}, m_{22}, m_{24}, m_{25}\} - p_{123}\{a_7, a_8, a_{11}, a_{13}, a_{14}\} = 0,0923.$$

На основе анализа указанных результатов получена следующая ранжировка метрик сложности из состава метрического базиса:

$$\begin{aligned} & \{m_{11}, m_{15}, m_{21}, m_{23}\} \succ \\ & \{m_{12}, m_{13}, m_{14}, m_{16}, m_{19}\} \succ \\ & \{m_{17}, m_{18}, m_{22}, m_{24}, m_{25}\}. \end{aligned}$$

Таким образом, по результатам экспертного оценивания как наиболее предпочтительные для статического анализа выделены метрические характеристики длины текста, метрики Джилба и точек пересечения, а также метрика интеллектуального содержания, что вполне согласуется с представленными ниже результатами экспериментальных исследований.

Структурирование предложенного метрического базиса на основе экспериментального исследования чувствительности метрик сложности к дефектам безопасности

В зависимости от характеристик и особенностей метрик сложности им можно ставить в соответствие порядковую либо относительную измерительные шкалы. Порядковой шкале соответствуют метрики, позволяющие ранжировать некоторые характеристики путем сравнения с опорными (эталонными) значениями, т. е. измерение по этой шкале фактически определяет взаимное положение метрик для конкретных программ. Относительной шкале соответствуют метрики, позволяющие не только оценить рассчитанные значения относительно друг друга, но и определить, как далеко оценки отстоят от границы, начиная с которой характеристика может быть измерена [2, 11].

Оправдан подход к расчету значений метрики сложности как для каждого отдельного функционального объекта (ФО) или модуля в составе исследуемого программного проекта, так и значения метрики для программного проекта в целом [3, 9]. В данном случае наибольшее внимание необходимо уделить не столько отдельным конкретным значениям вычисляемых метрик, сколько соотношению

их значений друг с другом. Для оценивания характера относительных изменений метрик информационной сложности были рассчитаны метрические показатели для трех модификаций программы, различающихся способом внедрения в исходный код тестовых потенциально опасных конструкций.

ПОК можно вносить в ручном или в автоматизированном режиме. Автоматический режим создания ПОК подразумевает использование ранее объявленных в программе операторов и операндов. После внедрения ПОК в автоматизированном режиме при неизменном словаре программы значительно увеличивается средний размер кода программы. Вследствие появления такой большой избыточности значение метрики интеллектуального содержания ощутимо увеличивается, а значение уровня реализации программы многократно уменьшается (более чем в 10 раз), что также свойственно вирусной технологии.

В ручном режиме можно максимально замаскировать ПОК путем имитации методов построения программы, примененных разработчиком, и внесения минимально возможных изменений в набор маршрутов выполнения программы. После внедрения такой ПОК метрика уровня реализации практически соответствует исходной программе. Словарь программы при этом незначительно расширяется, что объясняется применением небольшого количества дополнительных операторов, а среднее значение метрики интеллектуального содержания значительно увеличивается.

Таким образом, ясно прослеживается зависимость интеллектуального содержания программы от наличия в ней ПОК. Это объясняется двумя причинами: наличием у внедряемого фрагмента кода собственного интеллектуального содержания и прямой зависимостью между объемом программы и ее интеллектуальным содержанием. Следовательно, наиболее чувствительным показателем наличия в программе внедренного кода является метрика, характеризующая интеллектуальное (информационное) содержание программы.

То есть наблюдаемые относительные изменения значений метрик инвариантны к размеру исследуемой программы или отдельного участка кода. Следовательно, сравнение и анализ рассчитанных для отдельных участков кода и/или функциональных объектов значений выбранной взаимоувязанной тройки метрик, особенно метрики интеллектуального содержания, позволяют более точно локализовать и в итоге – выявлять потенциально опасные программные конструкции.

Для формирования критерия распознавания признаков ДБ, основанного на сложности управляющего графа программы, помимо абсолютных значений метрик оценивались относительные отклонения по участкам кода, математическое ожидание и среднеквадратическое отклонение в рамках исследуемой программы. В примере в табл. 2 наибольшие отклонения показаны у ФО под номерами 5,

6, 7. Следовательно, потребуется более углубленное изучение данных функциональных объектов как предположительно содержащих в себе ПОК.

В целом экспериментальные исследования выбранных метрик сложности показали высокий уровень их корреляции (0,93–0,98) с общим количеством дефектов.

Метрики сложности из сформированного метрического базиса можно рассчитывать уже на первом этапе статического анализа. Но необходимо учитывать, что первоначальное состояние программного средства и составляющих его ФО, как правило, достаточно обфускировано. В связи с этим рассчитанные значения метрик сложности не всегда достоверно отражают удовлетворение критериям. Тем не менее, получение значений метрик на данном этапе и их сравнительный анализ в качестве экспресс-исследования

ТАБЛИЦА 2. Исследование программного проекта с разделением по функциональным объектам

Объект	Словарь n	Уровень реализации L	Интеллектуальное содержание I	Метрика Джилба f^*	Метрика точек пересечения
ФО 1	21	0,070	44,982	2,843	5586
ФО 2	22	0,021	54,784	2,825	5585
ФО 3	17	0,118	61,962	2,869	5580
ФО 4	21	0,104	62,780	2,849	5589
ФО 5	21	0,017	48,538	2,772	5578
ФО 6	17	0,222	94,235	2,894	5593
ФО 7	18	0,222	87,538	2,872	5575
ФО 8	23	0,025	69,651	2,828	5583
ФО 9	19	0,154	67,556	2,779	5586
ФО 10	16	0,182	71,636	2,793	5589
ФО 11	15	0,118	67,556	2,791	5589
Математическое ожидание	19,091	0,114	66,474	2,831	5585,435
Среднеквадратичное отклонение	2,663	0,076	14,859	0,041	5,017

позволяет выявлять потенциально наиболее опасные ФО.

Для получения более полной и достоверной информации необходимо рассчитать значения метрики Джилба и метрики точек пересечения после очистки программы, т. е. после устранения дефектов, выражающихся, в первую очередь, зацикливанием и избыточностью. Вычисление значений метрик сложности по каждому ФО после таких преобразований дает возможность наиболее достоверно оценить информационные и топологические характеристики программы и как следствие – более точно локализовать потенциально опасные конструкции.

На основе полученных данных сформулированы критерии для классификации выявленных ДБ как потенциально опасных программных конструкций:

- характер относительных изменений между отдельными ФО и ПС в целом метрик информационной сложности: теоретической и реальной длины программы, метрики интеллектуального содержания и ее составляющих;
- превышение для метрик Джилба и точек пересечения, усредненного по ФО значения отношений среднеквадратического отклонения к математическому ожиданию, некоторых граничных значений, устанавливаемых экспериментально.

Заключение

Рассмотрены два взаимодополняющих направления формирования информативного структурированного метрического базиса. Приведен вариант ранжирования метрик сложности на основе экспертного оценивания для выявления дефектов безопасности программного обеспечения. Описаны результаты экспериментального исследования чувствительности компонент расширенного метрического базиса к потенциально опасным программным конструкциям. Предлагаемые варианты комбинаций метрик сложности могут использоваться как дополнительные

инструменты при сертификационных испытаниях программных средств по требованиям безопасности информации.

Библиографический список

1. Ганцева Е. А. Статистический подход разрешения неопределённости экспертных суждений на основе теории случайных множеств / Е. А. Ганцева, В. А. Каладзе // Вестн. ВГУ. Сер. Системный анализ и информационные технологии. – 2015. – № 2. – С. 82–88.
2. Диасамидзе С. В. Использование метрик сложности для оценивания качества программ / С. В. Диасамидзе, В. А. Новиков, А. А. Платонов // Информационные технологии на железнодорожном транспорте : докл. XII междунар. науч.-практич. конф. «Инфотранс-2007». – СПб. : ПГУПС, 2008. – С. 70–72.
3. Диасамидзе С. В. Метод выявления недеklarированных возможностей программ с использованием структурированных метрик сложности : автореф. дис. ... канд. техн. наук : 05.13.19 / С. В. Диасамидзе : Петербург. гос. ун-т путей сообщения. – СПб., 2012. – 16 с.
4. Диасамидзе С. В. Метод и методика выявления недеklarированных возможностей программ с использованием структурированных метрик сложности / С. В. Диасамидзе // Изв. ПГУПС. – 2012. – Вып. 3 (32). – С. 29–36.
5. Коваленко И. И. Оценка качества программных продуктов с использованием теории Демпстера – Шейфера / И. И. Коваленко, А. В. Швед // Наукові праці. Серія : Комп'ютерні технології. – Т. 160, вип. 148. – Миколаїв : Редакційно-видавничий центр ЧДУ ім. Петра Могили, 2011. – С. 22–26.
6. Корниенко А. А. Информационная безопасность и защита информации на железнодорожном транспорте: в 2 ч. Ч. 2. Программно-аппаратные средства обеспечения информационной безопасности на железнодорожном транспорте / А. А. Корниенко и др. ; под ред. А. А. Корниенко. – М. : УМЦ ЖДТ, 2014. – 448 с.
7. Корниенко А. А. Методическое и инструментальное обеспечение оценки соответствия программных средств железнодорожного транс-

порта в системе требований «качество – безопасность» / А. А. Корниенко, М. Л. Глухарев, С. В. Диасамидзе и др. // Интеллектуальные системы на транспорте : сб. материалов III междунар. науч.-практич. конф. «ИнтеллектТранс-2013». – СПб. : ПГУПС, 2013. – С. 80–89.

8. Корниенко А. А. Обязательное подтверждение соответствия программных средств железнодорожного транспорта / А. А. Корниенко, В. И. Сафонов // Транспорт РФ. Спецвыпуск «Наука и транспорт». – 2009. – № 3 (22). – С. 36–39.

9. Маевский Д. А. Оценка количества дефектов программного обеспечения на основе метрик сложности / Д. А. Маевский, С. А. Яремчук // Электротехнические и компьютерные системы. – 2012. – № 07 (83). – С. 113–120.

10. Майерс Г. Надежность программного обеспечения / Г. Майерс ; пер. с англ. ; под ред. В. Ш. Кауфмана. – М. : Мир, 1980. – 360 с.

11. Платонов А. А. Контроль целостности динамических объектов вычислительных систем с использованием метрических эталонов / А. А. Платонов, В. И. Тимофеев // Тр. СПИИРАН. – 2015. – Вып. 1 (38). – С. 136–160.

12. Саати Т. Л. Принятие решений. Метод анализа иерархий / Т. Л. Саати. – М. : Радио и связь, 1989. – 316 с.

13. Хованов Н. В. Анализ и синтез показателей при информационном дефиците / Н. В. Хованов. – СПб. : СПбГУ, 1996.

14. Холстед М. Х. Начала науки о программах / М. Х. Холстед ; пер. с англ. В. М. Юфы. – М. : Финансы и статистика, 1981. – 128 с.

15. Dempster A. P. A generalization of Bayesian inference / A. P. Dempster // J. R. Stat. Soc. – 1968. – Vol. 30. – P. 205–247.

16. McCabe T. J. A complexity measure // IEEE Trans. Software Eng. – 1976. – Vol. SE-2, N 4, Dec. – P. 308–320.

17. Neelamegam C. Enhanced ensemble prediction algorithms for detecting faulty modules in object oriented systems using quality metrics / C. Neelamegam, M. Punithavalli // J. Comp. Sci. – 2012. – Vol. 8, Is. 12. – P. 2075–2082.

18. Shafer G. A Mathematical Theory of Evidence / G. Shafer. – Princeton : Princeton Univ. Press, 1976. – 297 p.

References

1. Gantseva Ye. A. & Kaladze V. A. *Vestnik VGU. Seriya: Sistemnyy analiz i informatsionnyye tekhnologii – Annals of the Voronezh State University. Systemic Analysis and Information Technologies Series*, 2015, no. 2, pp. 82-88.

2. Diasamidze S. V., Novikov V. A. & Platonov A. A. Ispolzovaniye metrik slozhnosti dlya otsenivaniya kachestva programm [Using Complexity Metrics for Assessment of Quality of Programs]. *Informatsionnyye tekhnologii na zheleznodorozhnom transporte: doklady XII mezhdunarodnoy nauchno-prakticheskoy konferentsii „Infotrans-2007“ [Information Technologies In Rail Transport: Proc. of the 12th Int. Sci. and Pract. Conf. „Infotrans-2007“]*. St. Petersburg, PGUPS, 2008. Pp. 70-72.

3. Diasamidze S. V. Metod vyyavleniya nedeklarirovannykh vozmozhnostey program s ispolzovaniyem strukturirovannykh metrik slozhnosti: avtoreferat dissertatsii ... kand. tekhn. nauk [A Method for Identification of Undocumented Features of Programs with the Use of Structured Complexity Metrics: Extended Abstract of a Thesis for the Degree of Cand. Sci. (Eng.)]. Petersburg State Transp. Univ. St. Petersburg, 2012. 16 p.

4. Diasamidze S. V. *Izvestiya PGUPS – Proc. Petersburg Transp. Univ.*, 2012, Is. 3 (32), pp. 29-36.

5. Kovalenko I. I. & Shved A. V. Otsenka kachestva programnykh produktov s ispolzovaniyem teorii Dempstera – Sheyfera [Assessment of Quality of Programming Products with the Use of the Dempster – Shafer Theory]. *Naukovi pratsi. Seriya: Kompyuterni tekhnolohiyi [Proc. Computer Technologies Series]*. T. 160, Is. 148. Mykolaiv: Redaktsiyno-vydavnychy tsentr ChDU im. Petra Mohyly, 2011. Pp. 22-26.

6. Korniyenko A. A. et al. Informatsionnaya bezopasnost i zashchita informatsii na zheleznodorozhnom transporte [Information Security and Data Protection in Rail Transport], 2 Vols. Vol. 2. Programmno-apparatnyye sredstva obespecheniya informatsionnoy bezopasnosti na zheleznodorozhnom transporte [Software and Hardware Tools for Ensuring Information Security in Rail Transport]; ed. A. A. Korniyenko. Moscow, UMTs ZhDT, 2014. 448 p.

7. Korniyenko A. A., Glukharev M. L., Diasamidze S. V., Zakharchenko S. S. & Polyanchko M. A.

Metodicheskoye i instrumentalnoye obespecheniye otsenki sootvetstviya programmnykh sredstv zheleznodorozhnogo transporta v sisteme trebovaniy «kachestvo – bezopasnost» [Methodological and Tool Support for Assessment of Correspondence of Rail Transport Software in the Quality – Safety Requirements System]. *Intellektualnyye sistemy na transporte: sbornik materialov III nauchno-prakticheskoy konferentsii “IntellektTrans-2013” [Intellectual Systems on Transport: Coll. Papers of the 3rd Sci. and Practical Conf. IntellektTrans-2013]*. St. Petersburg, PGUPS, 2013. Pp. 80-89.

8. Korniyenko A.A. & Safonov V.I. *Transport Rossiyskoy Federatsii. Spetsvyypusk “Nauka i Transport” – Transport of the Russian Federation. Sci. and Transport Spec. Is.*, 2009, no. 3 (22), pp. 36-39.

9. Mayevskiy D.A. & Yaremchuk S.A. *Elektrotekhnicheskkiye i kompyuternyye sistemy – Electro-technical and Computer Systems*, 2012, no. 07 (83), pp. 113-120.

10. Mayers G. Nadezhnost programmogo obespecheniya [Software Reliability]. ed. V.Sh. Kaufman. Moscow, Mir, 1980. 360 p.

11. Platonov A.A. & Timofeyev V.I. Kontrol tselostnosti dinamicheskikh obyektov vychislitelnykh sistem s ispolzovaniyem metricheskikh etalonov

[Continuity Test of Dynamic Objects of Computing Systems Using Metrical Templates]. *Trudy SPIIRAN – Proc. of St. Petersburg Inst. for Informatics and Automation of the Russian Acad. of Sci.*, 2015, Is. 1 (38), pp. 136-160.

12. Saati T.L. Prinyatiye resheniy. Metod analiza iyerarkhiy [Decision Making. Analytic Hierarchy Process]. Moscow, Radio i svyaz, 1989. 316 p.

13. Khovanov N.V. Analiz i sintez pokazateley pri informatsionnom defitsite [Analysis and Synthesis of Indices under Information Scarcity]. St. Petersburg, SPbGU, 1996.

14. Kholsted M.Kh. Nachala nauki o programmakh [Programming Science Basics]. Moscow, Finansy i statistika, 1981. 128 p.

15. Dempster A.P. A generalization of Bayesian inference. *J.R. Stat. Soc.*, 1968, Vol. 30, pp. 205-247.

16. McCabe T.J. A complexity measure. *IEEE Trans. Software Eng.*, 1976, Vol. SE-2, no. 4, Dec., pp. 308-320.

17. Neelamegam C. & Punithavalli M. Enhanced ensemble prediction algorithms for detecting faulty modules in object oriented systems using quality metrics. *J. Comp. Sci.*, 2012, Vol. 8, Is. 12, pp. 2075-2082.

18. Shafer G. A Mathematical Theory of Evidence. Princeton, Princeton Univ. Press, 1976. 297 p.

КОРНИЕНКО Анатолий Адамович – доктор техн. наук, профессор, заведующий кафедрой, каа. pgups@ya.ru; *ДИАСАМИДЗЕ Светлана Владимировна – канд. техн. наук, sv.diass99@ya.ru (Петербургский государственный университет путей сообщения Императора Александра I).