УДК 004.43
DOI 10.26732/2618-7957-2018-1-47-52

МЕТОДИЧЕСКИЙ ПОДХОД К ОЦЕНКЕ КОЛИЧЕСТВА НЕОБНАРУЖЕННЫХ ДЕФЕКТОВ ЕСТЕСТВЕННОЙ СЕМАНТИКИ ПРОГРАММЫ С ТРЕБУЕМОЙ СТЕПЕНЬЮ ДОВЕРИЯ ПРИ ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ БОРТОВЫХ ЦИФРОВЫХ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСОВ КОСМИЧЕСКИХ АППАРАТОВ

Д. С. Викторов, Е. Н. Жидков, Р. Е. Жидков
Военная академия воздушно-космической обороны имени Маршала Советского Союза Г. К. Жукова, г. Тверь, Российская Федерация

В рамках создания метода верификации программного обеспечения бортовых цифровых вычислительных комплексов космических аппаратов, удовлетворяющего задачам процесса верификации и разрабатываемого с целью снижения суммарных затрат на верификацию, представлена методика оценки количества не выявленных дефектов естественной семантики программы. Обнаружение данного типа дефектов производится путем статического анализа исходного кода программы и основывается на контроле соблюдения принципа размерной однородности выражений. Оценка количества дефектов выполняется по модели надежности Миллса, относящей к классу статистических моделей с преднамеренным внесением дефектов в программное обеспечение. Процедура внесения дефектов демонстрируется на теоретико-множественном представлении программы с учетом характерных для дефектов естественной семантики программы особенностей исходного кода, влияющих на адекватность проводимой процедуры. Требуемая степень доверия к результатам оценки необнаруженных дефектов достигается за счет вычисления математического ожидания количества обнаруживаемых «преднамеренных» дефектов по выборке требуемого размера, которая зависит от статистических характеристик, получаемых из начальной выборки и заданных значений доверительной вероятности и доверительного интервала. Полученная методика может быть использована при проведении испытаний исходного кода программ в рамках процесса верификации программного обеспечения бортовых цифровых вычислительных комплексов космических аппаратов, так как основана на общеизвестных математических соотношениях и соответствует требованиям нормативных документов в данной области.

Ключевые слова: программное обеспечение, бортовой цифровой вычислительный комплекс, космический аппарат, верификация, статический анализ, внесение дефектов, естественная семантика.

Постоянное повышение требований к разрабатываемым бортовым цифровым вычислительным комплексам космических аппаратов (БЦВС КА) вызвало рост сложности и размера программного обеспечения (ПО) как наиболее гибкого компонента, расширяющего функциональные возможности КА. Данный факт обусловил значительное увеличение трудозатрат на разработку ПО БЦВС КА как в абсолютном значении, так и в сравнении с трудозатратами на создание технических средств. По некоторым оценкам, 15 % от общего количества затрат отводится на верификацию ПО, при этом наблюдается тенденция к увеличению этой доли. В условиях секвестра бюджета из-за сложной внешнеэкономической ситуации вокруг России важным является использование таких методов разработки ПО, в том числе методов верификации, которые позволяют экономить государственные средства [1, 2].

Основной путь снижения затрат на верификацию ПО – максимальная автоматизация процесса исследования программных продуктов.

Методом верификации, имеющим наибольшую степень автоматизации, служит статический анализ (СА). Главный недостаток СА – ограниченное множество типов обнаруживаемых дефектов ПО, расширение которого является основным направлением развития данной группы методов верификации ПО [3–5].

Для снижения трудозатрат на верификацию ПО предложен подход, основанный на СА программ на предмет обнаружения дефектов естественной семантики (ДЕС). Естественная семантика представляет собой размерность физических величин, интерпретируемых в идентификаторах исходного кода (ИК) программы. Для реализации контроля отсутствия ДЕС необходимо выполнить отображение множества идентификаторов на множество векторов, описывающих размерность, а также отобразить операции с идентификаторами на операции с векторами. Далее производятся вычисления с векторами в порядке, описываемом абстрактным синтаксическим деревом с контролем условий корректности.

Структура метода верификации ПО основывается на задачах, стоящих перед данным интегральным процессом разработки. Наравне с обнаружением и регистрацией программных дефектов в рамках верификации ПО встроенных систем необходимо доказать устранение дефектов с высокой степенью доверия, под которым предлагается понимать конкретные значения доверительного интервала $I_{\beta}^{\text{ТРЕБ}}$ и доверительной вероятности $\beta_{\text{ТРЕБ}}$ [6–8].

Для доказательства устранения дефектов ПО воспользуемся величиной, характеризующей количество необнаруженных дефектов:

$$N_{\text{НЕОБЩ}} = N_{\text{ОБЩ}} - N_{\text{ОБН}}, \qquad (1)$$

где $N_{\text{ОБЩ}}$ – количество «собственных» дефектов программы до проведения верификации ПО; $N_{\text{ОБН}}$ – количество обнаруженных «собственных» дефектов программы.

На практике для оценки количества дефектов до начала верификации $N_{\text{ОБЩ}}$ применяются различные модели надежности: эвристическая, Джелински-Моранды, Миллса.

Простая эвристическая модель и модель Джелински-Моранды не могут применяться для оценки количества ДЕС, так как идеи, лежащие в их основе, противоречат сути СА, позволяющего по некоторой математической модели обнаруживать все возможные дефекты за один проход анализатора независимо от того, каким специалистом осуществляется исследование. При этом если рассматривать СА как контрольный пример в рамках тестирования, то интенсивность появления дефектов после его выполнения снизится сразу до нуля, что не позволит анализировать временные интервалы между проявлениями дефектов [9, 10].

Учесть вероятностный характер выявления ДЕС в оценке их количества до начала верификации, обусловленный характеристиками ИК программы и неизвестным законом распределения дефектов по составляющим выражений кода, возможно с использованием статистических моделей надежности, основанных на преднамеренном внесении дефектов в ПО. К классу подобных моделей относится модель Миллса, в которой вероятности обнаружения «собственных» дефектов и преднамеренно внесенных считают равными, а выражение для количества дефектов в программе до начала испытаний, исходя из принципа максимального правдоподобия, имеет вид [10]

$$N_{\text{ОБЩ}} = \frac{N_{\text{ОБЩ}}^{\text{ПР}} N_{\text{ОБН}}}{N_{\text{ОБН}}^{\text{ПР}}}, \qquad (2)$$

где $N_{\text{ОБЩ}}^{\text{ПР}}$ – количество преднамеренно внесенных дефектов в программу; $N_{\text{ОБН}}^{\text{ПР}}$ – количество обнаруженных дефектов программы из числа преднамеренно внесенных.

В модели надежности Миллса преднамеренное внесение дефектов производится до начала испытаний, что не может быть применено для оценки количества ДЕС в ПО, так как при этом существует вероятность взаимной компенсации с «собственными» дефектами программы. Свести риск подобной компенсации до нуля невозможно, однако минимальных значений для конкретного варианта ИК программы он достигнет при внесении дефектов после выявления «собственных». При этом $N_{\text{ОБН}}$ определяется методикой СА для поиска ДЕС программы и зависит от характеристик ИК программы, $N_{\text{ОБЩ}}^{\text{ПР}}$ обусловливается количеством операторов программы, содержащих операции с естественной семантикой, $N_{\text{ОБН}}^{\text{ПР}}$ в случае многократного проведения опытов будет иметь различные значения, что объясняется случайным попаданием преднамеренно вносимого ДЕС в одну из составляющих оператора (операцию или операнд). Закон распределения $N_{\text{ОБН}}^{\text{ПР}}$ неизвестен, поэтому в соотношении (2) предлагается использовать оценку математического ожидания (МОЖ) данной величины $\tilde{M}\left[N_{\text{ОБН}}^{\text{ПР}}\right]$:

$$N_{\text{ОБЩ}} = \frac{N_{\text{ОБЩ}}^{\text{ПР}} N_{\text{ОБН}}}{\tilde{M}[N_{\text{ОБН}}^{\text{ПР}}]}. \qquad (3)$$

Конечное выражение для количества необнаруженных дефектов получается путем замены в выражении (1) $N_{\text{ОБЩ}}$ на соотношение (3):

$$N_{\text{НЕОБН}} = N_{\text{ОБН}} \left(\frac{N_{\text{ОБЩ}}^{\text{ПР}}}{\tilde{M}[N_{\text{ОБН}}^{\text{ПР}}]} - 1 \right). \qquad (4)$$

Слабым местом моделей надежности, основанных на внесении дефектов, является процедура преднамеренного внесения, поскольку предполагается, что вероятности обнаружения «собственных» и преднамеренно внесённых дефектов равны [9]. Данная проблема возникает вследствие высокой сложности учета индивидуальных особенностей программистов (стиль, квалификация и т.п.) при выполнении прогноза возможной локализации «собственных» дефектов, который может быть использован в качестве множества потенциальных точек для искажения. Помимо своеобразия программистов необходимо уделять внимание и специфике характеристик ИК программы (для ДЕС это частота появления операндов), что является вполне формализуемой задачей.

Процесс внесения ДЭС предлагается рассматривать на представлении программы в виде мультимножества [11]:

$$SW = \{stmt_i \mid stmt_i \in STMT\}, i = \overline{1,I}, \qquad (5)$$

где $stmt_i \in STMT$ – i-й оператор программы, принадлежащий множеству операторов программы $STMT$.

Если абстрагироваться от синтаксиса конкретного языка программирования (ЯП), то любой оператор из (5), указывающий ЭВМ на выполнение вычисления, является некоторой комбинацией множества операндов, представленных идентификаторами ИК программы, и операций, определенных в спецификации ЯП:

$$stmt_i = \left(OPR_i^{\text{А}}, opr_i^{\text{П}}, opr_i^{\text{С}}, OPRND_i\right), \qquad (6)$$

где $OPR_i^{\text{А}} = \{opr_{ij}^{\text{А}} \mid opr_{ij}^{\text{А}} \in OPR_{\text{А}}, j = \overline{1,J}\}$ – мультимножество арифметических операций i-го оператора, порождаемое множеством арифметических операций ЯП $OPR_{\text{А}}$; $J = \left|OPR_i^{\text{А}}\right|$ – величина, характеризующая количество арифметических операций i-го оператора; $opr_i^{\text{П}} \in OPR_{\text{П}}$ – операция присваивания i-го оператора, принадлежащая множеству операций присваивания ЯП; $opr_i^{\text{С}} \in OPR_{\text{С}}$ – операция сравнения i-го оператора, принадлежащая множеству операций сравнения ЯП; $OPRND_i = \{oprnd_{il} \mid oprnd_{il} \in OPRND, l = \overline{1,L}\}$ – мультимножество операндов i-го оператора, порождаемое множеством операндов программы $OPRND$; $L = \left|OPRND_i\right|$ – величина, характеризующая количество операндов i-го оператора.

Различные варианты структуры оператора программы, используемые для внесения ДЕС, описываются на основе выражения (6):

$stmt_i = (OPR_i^{\text{А}}, opr_i^{\text{П}}, \varnothing, OPRND_i)$ – оператор, содержащий арифметические операции и операцию присваивания;

$stmt_i = (\varnothing, opr_i^{\text{П}}, \varnothing, OPRND_i)$ – оператор, содержащий операцию присваивания;

$stmt_i = (\varnothing, \varnothing, opr_i^{\text{С}}, OPRND_i)$ – оператор, содержащий операцию сравнения.

Характеристиками ИК программы, оказывающими влияние на эффективность поиска ДЕС, являются частоты появления операндов, имеющих одинаковые семантики. Очевидно, что чем больше переменных одной семантики в программе, тем выше должна быть вероятность их появления в качестве искажающего воздействия. Реализация данной закономерности осуществляется на мультимножестве, состоящем из Q вхождений операндов в программу и порождаемом множеством уникальных операндов:

$$OPRND_{BX} = \{oprnd_q \mid oprnd_q \in OPRND, q = \overline{1,Q}\}. \qquad (7)$$

Для выполнения выбора элемента set из множества SET по закону равномерной плотности распределения используется функция $rnd : SET \to set$. Реализация функции следующая:

$$rnd(SET) = set, \text{ при } P_{\text{ВЫБ}}(set) = \left(\left|SET\right|\right)^{-1}. \qquad (8)$$

Применяя функцию (8) к множеству (7). получаем элемент *oprnd* с учетом частоты появления операндов данной семантики в программе.

Процедура внесения ДЕС в программу рассматривается в рамках следующих допущений:

- внесение ДЕС осуществляется для каждого оператора программы только одиножды с целью исключения компенсации дефектов;
- внесение ДЕС производится для любого одного элемента оператора (операции или операнда) с одинаковой вероятностью;
- внесение ДЕС выполняется безотносительно к порядку выполнения операций с операндами.

Порядок выполнения процедур методики оценки количества необнаруженных ДЕС программы с требуемой степенью доверия при верификации ПО представлен на рисунке.

Входными данными для методики являются: ИК программа (*sw*); количество «собственных» ДЕС, обнаруженных при верификации ($N_{\text{ОБН}}$), доверительная вероятность требуемая ($\beta_{\text{ТРЕБ}}$), доверительный интервал требуемый ($I_{\beta}^{\text{ТРЕБ}}$), размер начальной выборки (K).

Процедура 1 – получение статистики.

Данная процедура разбивается на три шага: внесение ДЕС в программу, поиск ДЕС и возврат программы в исходное состояние.

Первый шаг начинается с внесения ДЕС в оператор программы $stmt_i$. Для чего выполняется поиск точки преднамеренного внесения

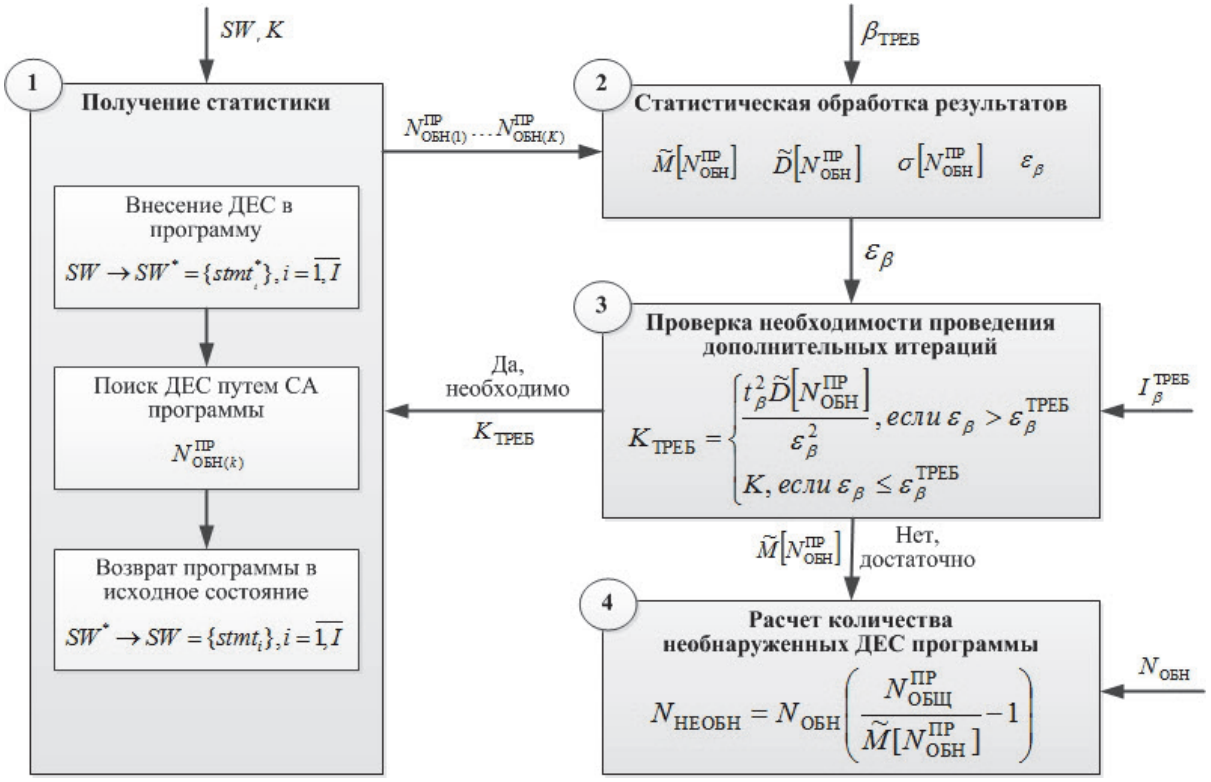

Рис. 1

дефекта pnt_i, которая выбирается из множества возможных точек для данного оператора

$$PNT_i = OPR_i^{\text{А}} \cup opr_i^{\text{П}} \cup opr_i^{\text{С}} \cup OPRND_i :$$
$$pnt_i = rnd(PNT_i).$$

Далее в зависимости от принадлежности pnt_i к одному из множеств

$$(OPR_{\text{А}}, OPR_{\text{П}}, OPR_{\text{С}}, OPRND_{BX})$$

выбирается один из элементов соответствующего множества, за исключением pnt_i, и используется в качестве дефекта: $def_i = rnd(SET \setminus pnt_i)$.

В результате вместо исходного оператора программы $stmt_i$ имеется $stmt_i^*$, у которого в одном из элементов описывающего его кортежа в точке pnt_i существует дефект def_i: $stmt_i \to stmt_i^*$. Выполнив внесение ДЕС для каждого оператора, получаем программу, в каждом операторе которой имеется дефект операции или операнда, распределённый равномерно:

$$SW \to SW^* = \{stmt_i^*\}, \quad i = \overline{1,I}.$$

На втором шаге производится обнаружение ДЕС с помощью методики СА для их поиска $N_{\text{ОБН}(k)}^{\text{ПР}}$.

Третий шаг заключается в возвращении программы в исходное состояние

$$SW^* \to SW = \{stmt_i\}, i = \overline{1,I}.$$

Шаги процедуры 1 выполняются K раз согласно заданному размеру начальной выборки.

Процедура 2 – статистическая обработка результатов.

Выполняется расчет следующих статистических характеристик [8]:

$\tilde{M}\left[N_{\text{ОБН}}^{\text{ПР}}\right] = K^{-1}\sum_{k=1}^{K} N_{\text{ОБН}(k)}^{\text{ПР}}$ – оценка МОЖ количества обнаруженных дефектов в серии из K итераций внесения дефектов;

$\tilde{D}\left[N_{\text{ОБН}}^{\text{ПР}}\right] = (K-1)^{-1}\sum_{k=1}^{K}\left(N_{\text{ОБН}(k)}^{\text{ПР}} - \tilde{M}[N_{\text{ОБН}}^{\text{И}}]\right)^2$ – оценка дисперсии количества обнаруженных дефектов в серии из K итераций внесения дефектов;

$\sigma\left[N_{\text{ОБН}}^{\text{ПР}}\right] = K^{-1}\sqrt{\tilde{D}[N_{\text{ОБН}}^{\text{ПР}}]}$ – среднее квадратическое отклонение оценки МОЖ количества обнаруженных дефектов в серии из K итераций внесения дефектов.

При допущении о нормальном распределении $N_{\text{ОБН}}^{\text{ПР}}$ ($K \ge 20\ldots30$) по таблице из [8] определяется значение величины t_β, соответствующее требуемому значению доверительной вероятности $\beta_{\text{ТРЕБ}}$.

Рассчитывается величина половины длины доверительного интервала для МОЖ количества

обнаруженных дефектов в серии из K итераций внесения дефектов: $\varepsilon_\beta = t_\beta \sigma[N_{\text{ОБН}}^{\text{ПР}}]$.

Процедура 3 – проверка необходимости проведения дополнительных итераций внесения дефектов.

Осуществляется сравнение значения ε_β с требуемым значением $\varepsilon_\beta^{\text{ТРЕБ}} = I_\beta^{\text{ТРЕБ}} / 2$.

Если $\varepsilon_\beta \le \varepsilon_\beta^{\text{ТРЕБ}}$, то рассчитанная оценка МОЖ количества обнаруженных «преднамеренных» дефектов считается совместной с данными, полученными в серии из K итераций внесения дефектов, согласно размеру начальной выборки.

Если $\varepsilon_\beta > \varepsilon_\beta^{\text{ТРЕБ}}$, то получен доверительный интервал для оценки МОЖ количества обнаруженных «преднамеренных» дефектов больше требуемого, что говорит о недостаточной репрезентативности рассматриваемой статистики дефектов и необходимости проведения дополнительных итераций внесения дефектов.

Для получения выборки требуемого размера $K_{\text{ТРЕБ}}$ используется система соотношений:

$$K_{\text{ТРЕБ}} = \begin{cases} t_\beta^2 \left(\varepsilon_\beta^2\right)^{-1} \tilde{D}\left[N_{\text{ОБН}}^{\text{ПР}}\right], \text{если } \varepsilon_\beta > \varepsilon_\beta^{\text{ТРЕБ}} \\ K, \text{если } \varepsilon_\beta \le \varepsilon_\beta^{\text{ТРЕБ}} \end{cases}.$$

Количество дополнительных итераций внесения дефектов рассчитывается по следующему соотношению: $K_{\text{ДОП}} = K_{\text{ТРЕБ}} - K$.

Если $K_{\text{ДОП}} > 0$, то необходимо вернуться к процедуре 1. Выполняется $K_{\text{ДОП}}$ итераций внесения дефектов и обрабатывается полученная статистика величиной $K_{\text{ТРЕБ}}$.

Процедура 4 – расчет количества необнаруженных ДЕС программы с требуемой степенью доверия.

Используя полученную оценку МОЖ количества обнаруженных «преднамеренных» дефектов для выборки требуемого размера $K_{\text{ТРЕБ}}$, при условии, что $N_{\text{ОБЩ}}^{\text{ПР}} = |SW|$, рассчитывается количество необнаруженных ДЕС программы по соотношению (4).

Представленный методический подход к оценке количества необнаруженных ДЕС программы с требуемой степенью доверия при верификации ПО БЦВС КА разработан с целью приведения в соответствие нормативным документам создаваемого метода верификации ПО и позволяет выполнять доказательство отсутствия дефектов рассматриваемого типа после выполнения их поиска путем СА с требуемой степенью доверия в рамках испытаний ИК программ.

Список литературы

1. Липаев В. В. Технико-экономическое обоснование проектов сложных программных средств. М. : СИНТЕГ, 2004. 284 с.
2. Мякишев Д. В. Принципы и методы создания надежного программного обеспечения АСУТП : метод. пособие. М. : Инфра – Инженерия, 2017. 114 с.
3. Кулямин В. В. Методы верификации программного обеспечения. М. : Институт системного программирования РАН, 2008. 117 с.
4. Карпов Ю. Г. Model checking. Верификация параллельных и распределённых программных систем. СПб. : БХВ-Петербург, 2010. 560 с. : ил.
5. Кларк Э. М., Грамберг О., Пелед Д. Верификация моделей программ : Model checking : пер. с англ. ; под ред. Р. Смолянского. М. : МЦНМО, 2002. 416 с. : ил.
6. ГОСТ Р 51904-2002. Программное обеспечение встроенных систем. Общие требования к разработке и документированию. М. : Госстандарт России, 2002. 94 с.
7. КТ-178. Квалификационные требования. Требования к программному обеспечению бортовой аппаратуры и систем сертификации авиационной техники. М. : Межгосударственный авиационный комитет, Авиационный регистр, 1996. 37 с.
8. Вентцель Е. С. Теория вероятностей : учебник для студ. вузов. 9-е изд., стер. М. : Издательский центр «Академия», 2003. 576 с.
9. Гуров Д. В., Гуров В. В., Иванов М. А. Использование моделей надежности программного обеспечения для оценки защищенности программного комплекса // Безопасность информационных технологий. 2012. № 1. С. 88–91.
10. Черников Б. В., Поклонов Б. Е. Оценка качества программного обеспечения: практикум : учеб. пособие ; под ред. Б. В. Черникова. М. : «ФОРУМ» ИНФРА-М, 2012. 400 с. : ил. (Высшее образование).
11. Петровский А. Б. Пространства множеств и мультимножеств. М. : Едиториал УРСС, 2003. 248 с.

История статьи
Поступила в редакцию 6 марта 2018 г.
Принята к публикации 5 апреля 2018 г.

METHODICAL APPROACH FOR EVALUATION OF THE NUMBER UNDETECTED NATURAL SEMANTICS DEFECTS WITH THE REQUIRED DEGREE OF CONFIDENCE IN THE VERIFICATION OF SPACE VEHICLES ON-BOARD COMPUTER SYSTEMS SOFTWARE

D. S. Viktorov, E. N. Zhidkov, R. E. Zhidkov
Military Aerospace Defense Academy, Tver, Russian Federation

The approach for evaluation the number of undetected natural semantics defects in the program is developed and presented in the framework of creating software verification method for space vehicles on-board computer systems. Method is created to reduce the total cost of verification that satisfied the tasks of the verification process. Detection of this defects type is performed within the static analysis of the program source code. It is based on monitoring compliance with the principle of dimensional uniformity of expressions. The estimation of the defects number is performed according by the Mills reliability model, which belongs to the statistical models class with deliberate injection of defects into the software. The defects injection procedure is demonstrated on the representation of the program via sets, taking into account the specific natural semantics defects characteristic of the source code, which affect the adequacy of the procedure performed. The required evaluation degree of confidence of undetected defects is achieved by calculating the number of detectable injected defects expected value. This statistic characteristic received from the selection of the required size, which depends on the statistical characteristics obtained from the initial selection and the given values of the confidence probability and the confidence interval. The obtained approach can be used in research of the programs source code within the process of software verification for space vehicles on-board computer systems, since it is based on well-known mathematical relationships and meets the requirements of normative documents in this field.

Keywords: software, space vehicle, on-board computer system, verification, static analysis, defects injections, natural semantics.

References

1. Lipaev V. V. *Tehniko-jekonomicheskoe obosnovanie proektov slozhnyh programmnyh sredstv* [Feasibility study of complex software projects]. Moscow, SINTEG Publ., 2004. 284 p.
2. Mjakishev D. V. *Principy i metody sozdanija nadezhnogo programmnogo obespechenija ASUTP* [Principles and methods of creating reliable software for the automated process control system]. Moscow, Infra – Inzhenerija Publ., 2017. 114 p.
3. Kuljamin V. V. *Metody verifikacii programmnogo obespechenija* [Methods of software verification]. Moscow, Institut sistemnogo programmirovanija RAN Publ., 2008. 117 p.
4. Karpov U. G. Model checking. *Verifikacija parallel'nyh i raspredelennyh programmnyh sistem* [Model checking. Verification of parallel and distributed software systems]. Saint Petersburg, BHV-Peterburg Publ., 2010. 560 p.
5. Clark E. M. *Verifikacija modelei programm* [Model checking verification]. Moscow, MCNMO, 2002. 416 p.
6. State standard 51904-2002. Software of embedded systems. General requirements for development and documentation. Moscow, Gosstandart Rossii Publ., 2002. 94 p.
7. Standard KT-178. Qualification requirements. Requirements for software on-board equipment and aviation equipment certification systems. Moscow, Mezhgosudarstvennyj aviacionnyj komitet, Aviacionnyj registr Publ., 1996. 37 p.
8. Ventcel' E. S. *Teorija verojatnostej* [Probability theory]. Moscow, Izdatel'skij centr "Akademija" Publ., 2003. 576 p.
9. Gurov D. V., Gurov V. V., Ivanov M. A. *Ispol'zovanie modelej nadezhnosti programmnogo obespecheniya dlya ocenki zashchishchennosti programmnogo kompleksa* [Use of software reliability models to assess the security of the software package]. Bezopasnost' informacionnyh tehnologij [Information Security], 2012, no. 1, pp. 88–91. (In Russian)
10. Chernikov B. V., Poklonov B. V. *Ocenka kachestva programmnogo obespechenija* [Software Quality Assessment]. Moscow, «FORUM» INFRA-M Publ., 2012. 400 p.
11. Petrovskij A.B. *Prostranstva mnozhestv i mul'timnozhestv* [Spaces of sets and multisets]. Moscow, Editorial URSS Publ., 2003. 248 p.

Article history
Received 6 March 2018
Accepted 5 April 2018