УДК 681.3.06:629.7.062

Ю.С. МАНЖОС

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

ОЦЕНКА ЭФФЕКТИВНОСТИ НЕЗАВИСИМОЙ ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Излагается методика оценки эффективности независимой верификации ПО информационно-управляющих систем критического применения на основе статического анализа исходных кодов ПО с использованием семантического метода оценки функциональности.

информационно-управляющие системы, независимая верификация, операционный спектр, семантический контроль, семантическое пространство, семантический спектр, сертификация ПО, экспертиза ПО

Введение

Безопасность информационно-управляющих систем (ИУС) авиационно-космическими объектами, АЭС существенно зависит от степени соответствия их программного обеспечения (ПО) регулирующим требованиям национальных и международных стандартов.

Основой сертификации и разрешительной деятельности являются экспертные оценки соответствия ПО ИУС регулирующим требованиям по безопасности и качеству. Существующая практика проведения экспертизы ПО основана на ручном анализе больших объемов проектной и нормативной документации, что обуславливает значительное влияние субъективных факторов на качество экспертных оценок. Как следствие существуют проблемы полноты и значительной трудоемкости экспертных оценок и связанные с ними риски существования невыявленных дефектов ПО [1, 2], являющихся источниками возможных отказов ИУС. Это обуславливает актуальность разработки и внедрения в практику экспертизы и сертификации ИУС компьютеризированных методов анализа и объективного инструментального оценивания ПО, основанных на использовании принципа диверсности (разнообразия), что обеспечит повышение полноты и достоверности экспертных оценок [3]. Одним из методов независимой верификации является предложенный автором семантический метод.

Суть метода заключается в следующем [4]. Вводится понятие семантического пространства, базисными векторами которого являются независимые единицы выбранной физической системы единиц, например СИ, СГС. Оценка качества ПО ИУС в семантическом пространстве осуществляется посредством контроля корректности семантических отображений, фактически реализованных программным кодом.

Важнейшей характеристикой качества, в соответствии со стандартом ISO/IEC 9126 2000 года, является **функциональность**.

Функциональность программного обеспечения оценивается совокупностью корректных семантических векторов операторных отображений семантики, фактически реализованных в оцениваемом программном продукте.

1. Формулирование проблемы

Для оценки эффективности независимой верификации необходимо знание проверяющей способности метода – вероятности обнаружения программного дефекта при условии его существования. В [5] рассматривалась оценка проверяющей способности, основанная на простой модели дефектов, предусматривающей только искажение операции в

исходном коде. Актуальным остается рассмотрение влияния семантического состава переменных программы (семантического спектра) на проверяющую способность метода.

Цель статьи – изучение семантического спектра (СС) реальных ИУС; влияние СС на проверяющую способность метода; оценка эффективности метода независимой верификации.

2. Решение проблемы

2.1. Семантический спектр ИУС критического применения

Для получения семантического спектра необходимо проанализировать реальные исходные коды ПО. Анализ ПО одной из подсистем реальной ИУС АЭС позволил получить следующее распределение программных переменных (табл. 1).

Таблица 1

Распределение переменных

Размерность	Кол. Переменных
безразмерные	142
м	135
с	5
Кельвин	142
Кельвин/м	2
Паскаль	302
Ватт	10
кг/с	21
$м^3/с$	32
м/с	16
оборот/с	9

Из табл. 1 видно, что доля переменных одной размерности лежит в диапазоне 0,25 … 37,5%. Среднее значение составляет 9%.

Таким образом, средние вероятности совпадения и различия физических типов (семантик) двух случайно выбранных программных переменных составляет $P_T = 0{,}09$, $P_{\overline{T}} = 0{,}91$.

2.2. Влияние семантического спектра ИУС на проверяющую способность семантического контроля

Для оценки проверяющей способности нам понадобятся данные об операционном спектре, приведенные в [6] (табл. 2).

Таблица 2

Операционный спектр

Операция	Вероятность
+	0,39
–	0,22
*	0,33
/	0,06

Операции сложения и вычитания будем называть аддитивными, а операции умножения и деления – мультипликативными. Тогда вероятности аддитивных и мультипликативных операций в исходном коде составляют:

$$P_A = 0{,}51;\ P_M = 0{,}391.$$

Рассмотрим простейшую модель дефектов, предполагающую, что в исходном коде может искажаться только операнд выражения, что может привести к изменению его типа в соответствии с семантическим спектром.

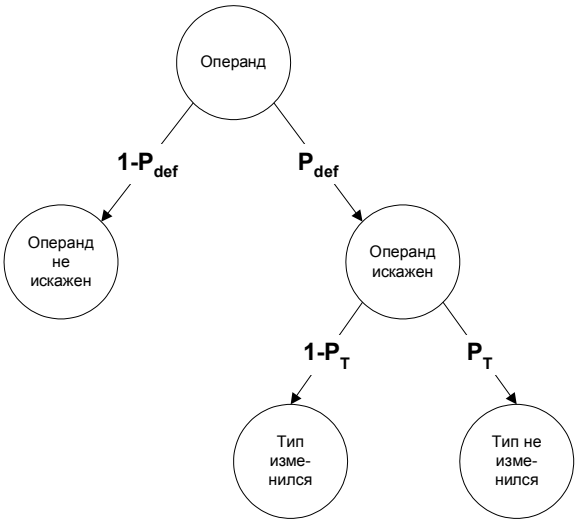

Рис. 1. Граф событий модели дефектов операнда:
P_{def} – вероятность возникновения дефекта;
P_T – вероятность сохранения типа операнда

Определим проверяющую способность метода как:

$$\eta = \frac{P_f}{P_f + P_{\overline{f}}}, \qquad (1)$$

где P_f – вероятность обнаружения дефекта;

$P_{\overline{f}}$ – вероятность необнаружения дефекта.

Вероятность обнаружения связана на графе (рис. 1) с дефектами, которые изменяют тип операнда:

$$P_f = P_{def}(1 - P_T), \quad (2)$$

где P_{def} – вероятность наличия дефекта;

P_T – вероятность сохранения типа операнда.

Вероятность необнаружения связана на графе (рис. 1) с дефектами, которые не изменяют тип операнда:

$$P_{\bar{f}} = P_{def} P_T. \quad (3)$$

Подставив (2), (3) в (1), получим, что проверяющая способность семантического контроля :

$$\eta = 1 - P_T. \quad (4)$$

Реальные значения семантического спектра показывают, что 90% случаев неправильного использования операндов (идентификаторов или указателей на переменные) может быть обнаружено семантическим контролем программного кода.

Рассмотрим более сложную модель дефектов, предполагающую, что искажению может быть подвергнут как один из операндов, так и бинарная операция. Кроме того, будем полагать, что в выражении возможен только один дефект. Такая модель позволит учесть влияние на проверяющую способность операционного и семантического спектров. Граф состояний полной модели дефектов приведен на рис. 2.

Для нахождения проверяющей способности полной модели дефектов (1), воспользуемся условием полной группы событий

$$P_n + P_f + P_{\bar{f}} = 1, \quad (5)$$

где P_n – вероятность неискажения выражения.

Подставив из (5) выражение для P_n в (1), получим:

$$\eta = \frac{P_f}{1 - P_n}. \quad (6)$$

На основании графа полной модели дефектов получим вероятность неискажения:

$$P_n = P_{\overline{def}} P_{\overline{def}} (P_+ P_{++} + P_- P_{--} + P_* P_{**} + P_/ P_{//}), \quad (7)$$

где $P_{\overline{def}}$ – вероятность отсутствия дефектов.

Заметим, что

$$P_+ + P_- + P_* + P_/ = 1. \quad (8)$$

Кроме того, для переходных вероятностей:

$$P_{++} + P_{+-} + P_{+*} + P_{+/} = 1; \quad (9)$$

$$P_{-+} + P_{--} + P_{-*} + P_{-/} = 1; \quad (10)$$

$$P_{*+} + P_{*-} + P_{**} + P_{*/} = 1; \quad (11)$$

$$P_{/+} + P_{/-} + P_{/*} + P_{//} = 1. \quad (12)$$

Примем вероятности искажения операций и операндов равными:

$$P_{+-} + P_{+*} + P_{+/} = P_{def}; \quad (13)$$

$$P_{-+} + P_{-*} + P_{-/} = P_{def}; \quad (14)$$

$$P_{*+} + P_{*-} + P_{*/} = P_{def}; \quad (15)$$

$$P_{/+} + P_{/-} + P_{/*} = P_{def}. \quad (16)$$

Предположив, что дефект имеет равномерное распределение по операциям, т.е.

$$P_{ij} = \text{const}. \quad (17)$$
$$i \neq j$$

На основании выражений (8) – (17) получим, что вероятность неискажения выражения равна

$$P_n = P_{\overline{def}}^3. \quad (18)$$

Найдем вероятность обнаружения дефекта:

$$P_f = P_{def} P_{\bar{T}} + P_{\overline{def}} P_{def} P_{\bar{T}} + P_+ (P_{+*} + P_{+/}) + P_- (P_{-*} + P_{-/}) + P_* P_{def} + P_/ P_{def}. \quad (19)$$

Подставив (18, 19) в (6) и воспользовавшись выражениями (8) – (17), получим

$$\eta = 1 - \frac{P_{\overline{def}}^2 \left(\frac{P_A}{3}\right) + P_{\overline{def}} P_T + P_T}{P_{\overline{def}}^2 + P_{\overline{def}} + 1}, \quad (20)$$

где $P_A = P_+ + P_-$ – вероятность аддитивных операций в коде;

$P_{\overline{def}}$ – вероятность отсутствия дефектов в коде;

P_T – вероятность сохранения типа при искажении операнда.

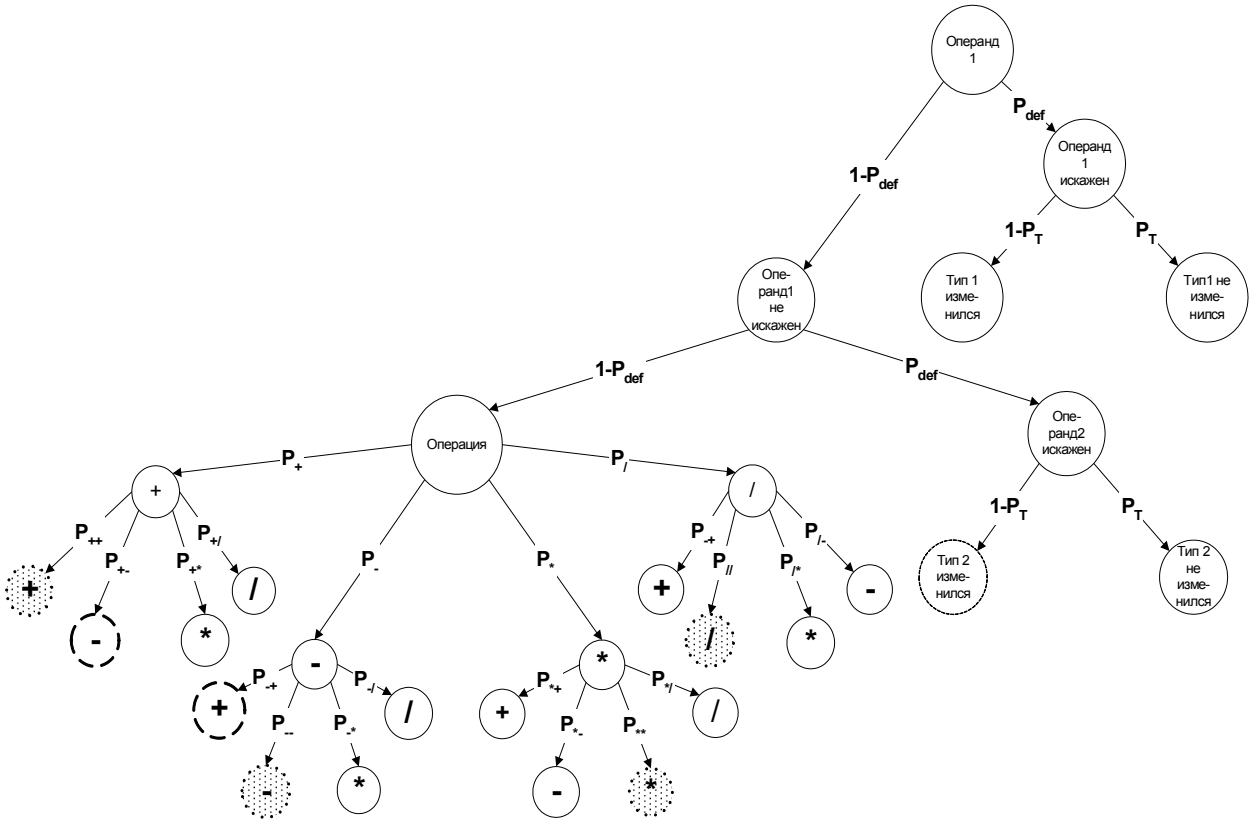

Рис. 2. Граф событий полной модели дефектов:
P_{def} – вероятность возникновения дефекта; P_T – вероятность сохранения типа операнда; P_+, P-, P_*, $P_/$ – операционный спектр; P_{ij} – переходные вероятности, определяющие искажение i-й операции в j-ю операцию; P_{ii} – вероятности, определяющие неискажение i-й операции

2.3. Оценка эффективности метода независимой верификации

Рассмотрим два крайних случая использования семантического контроля:

1. Начальная стадия разработки ПО, характеризующаяся очень большим количеством дефектов, $P_{\overline{def}} \to 0$ и проверяющей способностью:

$$\eta = 1 - P_T . \qquad (21)$$

2. Конечная стадия разработки ПО, характеризующая очень малым количеством дефектов, $P_{\overline{def}} \to 1$ и проверяющей способностью:

$$\eta = 1 - \frac{P_A}{9} - \frac{2P_T}{3} . \qquad (22)$$

Откуда, для $P_a = 0{,}51$, $P_т = 0{,}09$, $\eta = 0{,}88$.

Таким образом независимая верификация программного кода, основанная на его семантическом анализе, позволяет обнаружить в среднем 88% дефектов. Принимая во внимание возможность выполнения семантического контроля при статическом анализе, не связанном с исполнением программного кода, представляется возможным довести степень покрытия до 100%, что позволит выявить труднообнаруживаемые дефекты.

Заключение

Предлагаемый метод независимой верификации обладает высокой эффективностью, обеспечиваемой:

– *100% степенью покрытия* исходного кода;

– *разрешающей способностью,* ограниченной аддитивной операцией/операцией присваивания/операцией передачи параметров в процедуру.

Достоверность оценки – вероятность обнаружения программного дефекта, определяется статистическими характеристиками кода – операци-

онным и семантическим спектром и для реальных программных комплексов достигает 88%.

При использовании семантического контроля как одного из методов независимой верификации, общая достоверность оценки ПО значительно улучшится.

Диагностическая способность метода определяется моделью дефектов, учитывающей:

- несовпадение размерностей операндов аддитивных операций и операций присваивания;
- несовпадение размерностей при передаче параметров в алгоритмы;
- искажение знаков операций в арифметических выражениях.

Указанные категории дефектов не обнаруживаются существующими компиляторами, требуют значительных объемов тестирования обычными методами и являются причиной возможных скрытых дефектов ПО [7].

Представленный семантической метод оценки функциональности может быть использован для диверсификации технологий оценки качества ПО ИУС, важных для безопасности.

Использование метода позволяет уменьшить риски аномального функционирования ИУС и обеспечивает повышение общей безопасности применения ИУС.

Дальнейшим развитием разработанной инструментальной системы может быть создание экспериментальной базы нового поколения испытательных лабораторий и сертификационных центров, выполняющих экспертизу качества программных средств ИУС критического применения.

Литература

1. Манжос Ю.С. Методи підвищення якості програмного забезпечення // Міжнародна міждисциплінарна науково-практична конференція “Сучасні проблеми гуманізації та гармонізації управління”. Матеріали ММ НПК. – Х.: ХАІ. – 2001. – С. 134.

2. Манжос Ю.С. Методи підвищення якості програмного забезпечення РКТ // Міжнародна міждисциплінарна науково-практична конференція “Інтегровані комп’ютерні технології в машинобудуванні” ІКТМ-2001. Матеріали ММ НПК. – Х.: ХАІ, 2001. – С. 120.

3. Манжос Ю.С. Семантический контроль программного обеспечения систем критического применения // Авіаційно-космічна техніка і технологія. – Х.: НАКУ „ХАІ”. – 2002. – Вип. 34. – С. 207 – 212.

4. Манжос Ю.С. Принципы семантического контроля программного обеспечения систем критического применения // Авіаційно-космічна техніка і технологія. – Х.: НАКУ „ХАІ”. – 2002. – Вип. 32. – С. 307 – 315.

5. Харченко В.С., Манжос Ю.С., Петрик В.Л. Статистический анализ программного обеспечения систем управления космическим аппаратом и оценка проверяющей способности семантического контроля // Технология приборостроения. Научно-технический журнал. – Х.: ГП НИТИП, 2000. – № 2. – С. 52 – 59.

6. Харченко В.С., Шостак И.В., Манжос Ю.С. Принципы построения интеллектуальной системы сертификации программного обеспечения // Системи обробки інформації. – Х.: НАНУ, ПАНМ, ХВУ. – 2002. – Вип. 4 (20). – С. 3 – 7.

7. Манжос Ю.С. Типизация данных в системах критического применения // Системи обробки інформації. – Х.: НАНУ, ПАНМ, ХВУ. – 2002. – Вип. 3(19). – С. 54 – 57.

Поступила в редакцию 1.06.2004

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.