УДК 681.3.06

О.В. Боровик, доктор технических наук, профессор,
С.В. Ленков, доктор технических наук, профессор,
Р.П. Графов, кандидат технических наук, доцент

АНАЛИЗ ПАРАМЕТРОВ НАДЕЖНОСТИ СЛОЖНЫХ ПРОГРАММНЫХ КОМПЛЕКСОВ

В статье приведены модели и методы анализа и оценки надежности сложных программных комплексов, в основе которых использован стохастический ориентированный граф состояний программных модулей, образующих программный комплекс. Рассмотрены имеющие практическое значение задачи, позволяющие оценивать и обеспечивать требуемый уровень надежности программных комплексов в соответствии с предъявляемыми требованиями.

Ключевые слова: *сложный программный комплекс, программный модуль, стохастический ориентированный граф.*

У статті наведено моделі та методи аналізу й оцінювання складних програмних комплексів, в основу яких покладено використання стохастичного орієнтованого графа станів програмних модулів, що утворюють програмний комплекс. Розглянуто завдання, що мають практичне значення, які дозволяють оцінювати й забезпечувати бажаний рівень надійності програмних комплексів відповідно до висунутих вимог.

Ключові слова: *складний програмний комплекс, програмний модуль, стохастичний орієнтований граф.*

Models and methods of an analysis and estimation of the reliability of program complexes, based on the stochastic focused state graph of the program modules, forming a program complex, are resulted. Issues, having practical value and allowing estimating and providing the demanded level of reliability of program complexes according to the requirements, are considered.

Keywords: *program complex, program module, stochastic focused count.*

Настоящее время характеризуется интенсивным усложнением информационных технологий, масштабности прикладных задач в различных областях науки и техники, построением различных управляющих систем. Современное программное обеспечение (ПО) является неотъемлемой частью большинства систем управления. Его особенность состоит в том, что оно является достаточно сложным и крупномасштабным, а это повышает вероятность ошибок в программном коде и может привести к самым различным последствиям, включая катастрофические. Последнее накладывает дополнительные требования к обеспечению надежности ПО.

Особо остро данная проблема стоит в Вооруженных Силах, Службе безопасности, Министерстве Внутренних дел, Государственной пограничной службе Украины, других ведомствах, работающих в области национальной, в первую очередь информационной, безопасности государства.

Существует большое количество моделей и методов оценивания надежности программных средств (ПС), что вызвано наличием множества требований к функционированию ПС на различных этапах жизненного цикла. Поэтому задача оценивания надежности ПС пока не имеет общего и законченного решения [1, 2].

В соответствии с международным стандартом ISO 9126:1991 выделен ряд основных характеристик качества, позволяющих оценивать ПС. Среди них: надежность, корректность, удобство применения, эффективность, универсальность и сопровождаемость. Основным показателем качества является надежность, которая как динамический параметр проявляется на всем промежутке времени функционирования системы. Показатель надежности занимает особое место, так как в силу определения должен обеспечивать достаточно низкую вероятность отказа в процессе функционирования, быстрое реагирование на ошибки данных или вычислительного процесса и восстановление работоспособности, при котором не фиксируется отказ. Информационные системы функционируют в условиях действия внешних и внутренних факторов, способных нарушить их нормальную работу. Гарантоспособность как свойство системы должна противостоять этим факторам. Программное обеспечение является одной из главнейших составляющих системы, обеспечивающей ее гарантоспособность [3].

С учетом сказанного выше, задача оценивания и обеспечения заданной надежности сложных программных комплексов и проектов является актуальной в настоящее время. Целью работы является анализ моделей функционирования сложных программных комплексов и разработка методов оценивания и обеспечения параметров их надежности, имеющих практическое значение для проектирования и функционирования реальных сложных систем.

Для достижения цели в работе осуществлена постановка задач, позволяющих оценивать и обеспечивать требуемый уровень надежности программных комплексов в соответствии с предъявляемыми требованиями, а также проанализированы модели и методы их решения.

Одним из важнейших вопросов теории и практики надежности ПО является разработка математических моделей, методов и алгоритмов анализа расчета его надежности. Выбор оптимальной модели для оценки надежности ПО для практических целей на всем протяжении жизненного цикла является еще не до конца решенной задачей. Получение такой оценки необходимо и актуально для избежания всевозможных рисков (финансовых, временных) потерь, связанных с ошибками ПО, а также для эффективного планирования работы информационных систем, которые интенсивно разрабатываются и внедряются в наше время. Анализ надежности функционирования ПО осуществляется, как правило, до его сдачи в эксплуатацию на этапе отладки или тестирования. При этом для расчетов часто используется время, затрачиваемое на отладку и тестирование. Требования к указанным параметрам могут быть сформированы с учетом специфики и структуры ПО. Актуальность решения подобных задач вытекает из того, что ни один серьезный проект, связанный с решением комплекса важных и ответственных задач, не может быть сдан в эксплуатацию, без оценки надежности ПО на соответствие предъявляемым требованиям.

Под сложным программным комплексом (СПК) будем понимать сложную систему, содержащую множество независимых программных модулей (ПМ), функционирующих в соответствии с принципом построения СПК и своих собственных алгоритмов.

С учетом этого, задачи, позволяющие оценивать и обеспечивать требуемый уровень надежности программных комплексов в соответствии с предъявляемыми требованиями, можно представить в следующем виде.

1. Оценка надежности СПК, которая заключается в нахождении вероятности безошибочного функционирования СПК, при известной структурной схеме СПК и вероятности безошибочных решений задач всех ПМ.

2. Минимизация времени на отладку и тестирование при достижении заданного уровня надежности СПК.

3. Нахождение экстремальных значений показателя надежности СПК при ограничении времени на отладку и тестирование.

Модели и методы решения задачи

Программный комплекс СПК как сложная система содержит совокупность менее сложных составляющих ПМ, связанных между собой самым различным образом, определяемых структурой и спецификой СПК. Для исследования надежности СПК в целом предварительно рассмотрим модель надежности отдельных ПМ.

В настоящее время разработано множество моделей и методов оценивания надежности. Каждая из разработанных моделей оценки надежности ПМ основывается на системе допущений в зависимости от условий функционирования, среды эксплуатации, процессов корректировки программных ошибок. Из множества существующих моделей наиболее приемлемой для решения поставленных задач является экспоненциальная модель [4]. Данная модель учитывает время простоя и исправления ошибок. В процессе устранения ошибок эта модель учитывает реальное время работы ПМ. В соответствии с этой моделью вероятность безошибочной работы i-го ПМ можно представить следующей формулой:

$$P_i(t_i,\tau_i) = e^{\lambda_i t_i e^{-\nu_i \tau_i}}, \tag{1}$$

где λ_i – интенсивность появления ошибки при работе ПМ; ν_i – интенсивность отладки; t_i и τ_i – время вычислений и отладки модуля; $\lambda=1/T_i$; T_i – начальное среднее время безошибочной работы модуля; $=K_i/(N_{ошi}\ T_i)$; K_i – коэффициент сжатия времени отладки (тестирования) по сравнению с временем вычислений; $N_{ошi}$ – первоначальное (предполагаемое) число ошибок в ПМ.

Как видно, в выражении (1) для оценки надежности ПМ присутствуют также отладочные параметры и коэффициент *Ki*, позволяющий учесть уровень квалификации разработчика.

В качестве модели функционирования и оценки надежности СПК, содержащего множество независимых модулей, используем марковский процесс с дискретными состояниями и дискретным временем. Для такого процесса моменты $t_1, t_2, \ldots$, когда система S меняет свое состояние, рассматриваются как последовательность шагов, а в качестве аргумента, от которого зависит процесс, рассматривается номер шага: 1, 2, 3, ... Случайный процесс в этом случае характеризуется последовательностью состояний $S_0, S_2, \ldots, S_n$, которые образуют марковскую цепь. При этом S_0 – начальное состояние системы. Состояния цепи Маркова определяются вероятностями $P_i(k)$ вызова i-го программного модуля на k-ом шаге. Очевидно, что для любого k выполняется условие:

$$\sum_{i=1}^{n} P_i(k) = 1 \qquad (2)$$

Из состояния i в состояние j на k-ом шаге система переходит с вероятностями p_{ij}, которые образуют квадратную матрицу $|p_{ij}|$ размера (n x n), называемую стохастической матрицей вероятностей переходных состояний. Сумма переходных вероятностей в любой строке матрицы равна единице:

$$\sum_{i=1}^{n} p_{ij} = 1; (i = 1, \ldots, n). \qquad (3)$$

Используя свойства стохастической матрицы (2), можно определить вероятность состояния системы на к-м шаге по рекуррентной формуле:

$$P_i(k) = \sum_{j}^{n} P_j(k-1) p_{ji}; (i, j = 1, \ldots, n).$$

Рассмотрим простейший СПК, состоящий из N отдельных модулей, соединенных между собой в соответствии с алгоритмами обработки данных и решения задач. Пусть известны вероятности переходов между ПМ. Построим стохастический граф G(t), содержащий N+2 вершин (программных модулей). Начальная вершина N_0 означает начальное, а вершина N+1 − конечное состояние графа. Каждый ПМ решает свою задачу в соответствии с алгоритмом достижения целевой функции. Вершины 0-я и (N+1)-я являются фиктивными, время нахождения в них равно нулю, а вероятности безошибочной работы − единице. Фиктивные вершины вводятся в модель с целью корректного отображения начального состояния и динамики функционирования во времени всех ПМ. При этом предполагается, что модули статистически независимы. На рис. 1 показан стохастический граф с числом ПМ, равным N=5.

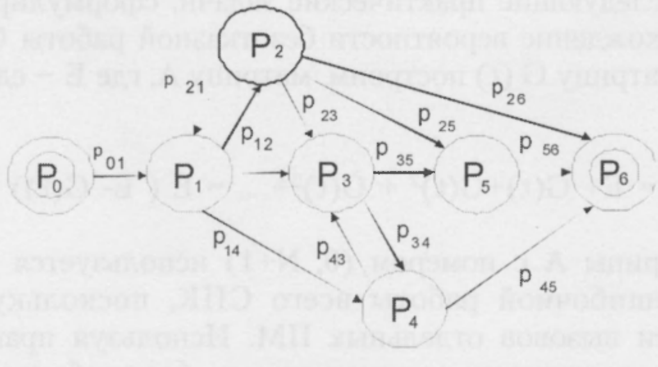

Рис. 1. Стохастический граф G(t)

Вершины графа − программные модули, а значения P (без аргумента для упрощения) − вероятности вызова ПМ, переходные вероятности p_{ij} определяют последовательность вызова программных модулей. В качестве элементов матрицы (см. рис. 1) графа G(t) используем произведения вероятностей перехода p_{ij} i-го

ПМ к j-му и вероятностей безотказного функционирования $P_i(t_i)$ i-го ПМ в течение времени t_i, (i,j = 0,...,(N+1)). Данное утверждение проистекает из того, что если система перешла из i–го состояния в состояние i+1, значит ПМ, соответствующий i-му состоянию, проработал нормально, без ошибок, в противном случае такой переход не состоялся бы.

В соответствии с приведенным графом G(t) построена стохастическая матрица вероятностей переходных состояний ПМ (рис. 2).

$$G(t) = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{12}P_1 & P_{13}P_1 & 0 & 0 & 0 \\ 0 & p_{21}P_2 & 0 & P_{23}P_2 & 0 & p_{25}P_2 & p_{26}P_2 \\ 0 & 0 & 0 & 0 & p_{34}P_3 & p_{35}P_3 & 0 \\ 0 & 0 & 0 & p_{43}P_4 & 0 & 0 & P_{46}P_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{56}P_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Рис. 2. Матрица вероятностей переходных состояний G(t)

Используя структурные преобразования над данной матрицей путем последовательного возведения ее в степень, можно найти вероятность безотказной работы комплекса за определенное количество шагов, включая последний шаг, на котором завершается процесс решения задачи. Чтобы найти указанную вероятность за два шага, нужно просуммировать произведения вероятностей по всем путям, соединяющим две вершины путем возведения матрицы в квадрат и т.д. Выполняя данную операцию над G при k $d \le k_{max}$, можно определить все связи между вершинами. Матрица $G^{k(\text{макс})}$ называется характеристической (матрицей всех путей между вершинами), где k_{max} такое, что дальнейшее возведение матрицы в степень не меняет ее вхождений, т.е. $k_{max} \le N-1$, так как максимальный ранг пути не может превышать значение N-1.

Рассмотрим следующие практические задачи, сформулированные ранее.

Задача 1. Нахождение вероятности безотказной работы СПК P(t).

Используя матрицу G (t) построим матрицу A, где E – единичная матрица [5, 6].

$$A = E + G(t) + G(t)^2 + G(t)^3 + \dots = E\,(E - G(t))^{-1}$$

Элемент матрицы A с номером (0, N+1) используется для определения вероятности безошибочной работы всего СПК, поскольку учитывает все последовательности вызовов отдельных ПМ. Используя правила вычисления обратной матрицы, выражение для вероятности безошибочной работы СПК с учетом всех ПМ можно представить в виде [6]:

$$P(t) = D(t) \,/\, R(t), \tag{4}$$

где: D(t) – алгебраическое дополнение элемента с номером (N+1, 0) матрицы (E – G(t)); R(t) – главный определитель матрицы (E-G(t)).

Выполнив преобразования в соответствие с (4) и предположив, что в выражении (1) для оценки надежности отдельных ПМ время отладки $\tau = 0$, получим искомую вероятность P(t).

Задача 2. Определение минимального общего времени отладки СПК, при котором $P(t, \tau) \geq P_{зад}$. Общее время отладки СПК определяется как:

$$\tau = \sum_{i=0}^{N+1} \tau_i.$$

Для решения данной и последующей задачи будем использовать метод неопределенных множителей Лагранжа [5]. Запишем функцию Лагранжа в виде:

$$f_1(t,\tau,\xi) = \sum_{i=0}^{N+1} \tau_i + \xi(P(t,\tau) - P_{зад}), \tag{5}$$

где ξ – множитель Лагранжа.

Дифференцируя (5) по τ и ξ и приравнивая полученные выражения нулю, получим систему уравнений, необходимую для дальнейших расчетов:

$$\begin{cases} \partial f_1(t,\tau,\xi) / \partial \tau_i = 0, \\ P(t,\tau) - P_{ад} = 0. \end{cases} \tag{6}$$

Искомое решение находится с учетом того, что $\tau = \tau^0$. Решив при этом второе уравнение системы (6), несложно получить

$$\tau^0 = \sum_{i=0}^{N+1} \tau^0,$$

Задача 3. Определение максимума вероятности безотказной работы комплекса $P(t, \tau)$ при заданном времени отладки $\tau_{зад}$. Запишем для этого случая функцию Лагранжа:

$$f_2(t,\tau,\omega) = P(t,\tau) + \omega(\sum_{i=0}^{N+1} \tau_i - \tau_{зад}) \tag{7}$$

Дифференцируя (7) по τ и ω и приравняв полученные выражения нулю, получим систему уравнений:

$$\begin{cases} \partial f_2(t,\tau,\omega) / \partial \tau_i = 0, \\ \sum_{i=0}^{N+1} \tau_i - \tau_{зад} = 0. \end{cases}$$

Решив второе уравнение приведенной системы, находим значения, используемые далее подстановкой для определения максимальной надежности СПК.

Рассмотренные в работе модели и методы позволяют производить анализ и оценку надежности сложных программных комплексов. Надежность составляющих модулей предварительно определяется с помощью метода, который в конкретном случае учитывает время на отладку и квалификацию специалистов, что в ряде случаев является важным на начальных этапах проектирования. Стохастическая матрица переходных состояний строится с использованием структуры и статистики алгоритма функционирования комплекса. Решение обратных задач позволяет производить расчет надежности с учетом ограничений на ресурсы, зависящих от квалификации разработчика программного обеспечения. Данный подход может быть использован для анализа и оценки надежности программных комплексов в целом, а также для исследования надежности комплекса на начальном этапе разработки, включающем этапы отладки и тестирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Липаев В.В.* Качество программного обеспечения / В.В. Липаев. – М. : Финансы и статистика, 1983. – 261 с.

2. *Карповский Е.Я.* Надежность программной продукции / Е.Я. Карповский, С.А. Чижов. – К. : Техника, 1990. – 160 с.

3. *Авижение А.Н.* Гарантоспособные вычисления : от идей до реализации в проектах / А.Н. Авиажение. – ТИИЭР, 1986. – Т. 74. – № 5 – С. 8–21.

4. *Musa J.D.* Validity of Execution time theory of software reliability // IEEE Trans. On reliability. – 1979. – № 3. – P. 199–205.

5. *Смагин В.А.* Метод оценивания и обеспечения надежности сложных программных комплексов. Техническое и метрологическое обеспечение, стандартизация и сертификация средств защиты информации / В.А. Смагин. – М. : Наука, 2005.

6. *Гантмахер Ф.Р.* Теория матриц / Ф.Р. Гантмахер. – М. : Наука, 1988. – 548 с.

Отримано 20.10.2011