

МОДЕЛИ ОЦЕНКИ НАДЕЖНОСТИ МОДУЛЬНЫХ СТРУКТУР N-ВАРИАНТНЫХ ПРОГРАММНЫХ СИСТЕМ ДОСТУПА К ШИРОКОПОЛОСНЫМ МУЛЬТИМЕДИЙНЫМ УСЛУГАМ

Ступина А. А.¹, Мельдер М. И.², Нургалева Ю. А.², Золотарев А. В.², Верхорубов А. И.²

¹ФГАОУ ВПО «Сибирский федеральный университет», Красноярск, Россия (660025, Красноярск, пер. Вузовский, 3)

²ФГБОУ ВПО «Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнёва», Красноярск, Россия (660014, Красноярск, пр. им. газ. «Красноярский рабочий», 31), e-mail: saa55@rambler.ru

В работе предлагаются модели оценки надежности модульных структур N-вариантных программных систем доступа к широкополосным мультимедийным услугам. Предлагается концепция COTS-сопровождения N-вариантного программирования. Она обеспечивает доступность математических моделей для оценки надежности версий модулей, что позволяет иметь информацию относительно как надежности ПО, так и его стоимости. Рассматриваются четыре типа базовых моделей оптимизации надежности программных систем (включая N-вариантные структуры) при условии ограниченности использования стоимостных ресурсов. Обобщающей моделью является модель В-2, формирующая оптимальный состав модулей многофункциональной (К-функций) N-вариантной программной системы с введением избыточности. По причине наличия ограничений и так как мы имеем дело с функциями системы ПО, ни один из методов, предложенных ранее, не может быть «напрямую» использован для решения задачи В-2. Вследствие нелинейности целевой функции в работе предложен метод линеаризации и представлены численные примеры решения задачи оптимизации состава версий модульных N-вариантных программных структур.

Ключевые слова: надёжное программное обеспечение, широкополосные мультимедийные услуги, N-вариантные программные системы.

ESTIMATION MODEL OF MODULAR STRUCTURES RELIABILITY OF THE N-VARIANT SOFTWARE SYSTEM ACCESS TO BROADBAND MULTIMEDIA SERVICES

Stupina A. A.¹, Melder M. I.², Nurgaleeva J. A.², Zolotarev A. V.², Vrhorubov A. I.²

¹Siberian Federal University, Krasnoyarsk, Russia (660025, Krasnoyarsk, Vuzovskii str., 3)

²Siberian State Aerospace University, Krasnoyarsk, Russia (660014, Krasnoyarsk, Krasnoyarskii rabochii av., 31), e-mail: saa55@rambler.ru

The paper suggests estimation model of modular structures reliability of the N-variant software system access to broadband multimedia services. The COTS-support concept of N-variant programming is offered. It ensures the availability of mathematical models to assess the reliability of the modules versions, that allows to get information about software reliability and cost. Let's consider four basic types of models of software systems optimization reliability (including N-variant structures), with the limited cost resources usage. A V-2 model is a general one, which forms the optimum module structure of multifunctional (K-functions) N-variant software system with the redundancy introduction. Due to the limitation presence and, as we deal with the functions of the system, none of the methods proposed earlier, cannot be directly used for the tasks B-2 solving. In consequence of the target function nonlinearity the paper proposes the linearization method and numerical examples of the optimization problem of versions of modular N-variant software structures solving.

Key words: reliable software, broadband multimedia services, N-variant software system.

С развитием сети Интернет правительства многих стран мира ощущают положительное влияние высокоскоростных сетей на коммерческие компании, общественные организации и рядовых граждан. Широкополосные сети давно стали неотъемлемой частью инфраструктуры мирового информационного сообщества. Они предоставляют пользователям постоянный высокоскоростной доступ к разнообразным веб-сервисам, контенту и программному обеспечению.

Недавние исследования доказали, что наличие надежных широкополосных сетей стимулирует рост ВВП промышленно развитых стран и может обеспечить аналогичные преимущества для стран с развивающейся рыночной экономикой. Поэтому возникает проблема надёжности программного обеспечения (ПО) технологии доступа к широкополосным мультимедийным услугам.

Предлагаются модели оценки надежности модульных структур N-вариантных программных систем, позволяющие алгоритмизировать оптимизационные процедуры, учитывающие фактор N-вариантности критического по отказоустойчивости ПО. Рассматриваются четыре типа базовых моделей оптимизации надежности программных систем (включая N-вариантные структуры) при ограничениях на использование доступных стоимостных ресурсов.

Предлагаемая концепция COTS-сопровождения N-вариантного программирования обеспечивает доступность математических моделей для оценки надежности версий модулей, что позволяет иметь информацию относительно как надежности ПО, так и его стоимости. Кроме того, следует отметить, что одно из главных преимуществ N-вариантного ПО – это независимость отказа избыточных версий модулей.

На основе концепции COTS-сопровождения N-вариантного программирования предлагается модель этапа постархитектуры для оценки стоимости N-вариантного ПО, которая используется в период, когда уже сформирована архитектура системы и выполняется дальнейшая разработка N-вариантных структур программного продукта. Основное уравнение постархитектурной модели является развитием уравнения модели СОСОМО II и имеет следующий вид:

$$COST = A \times K_{req} \times SIZE^B \times M_p + COST_{auto} \text{ [чел.-мес.]}$$

где: коэффициент K_{req} учитывает изменения в требованиях при формировании V-спецификаций на N-вариантные компоненты ПО;

показатель B отражает нелинейную зависимость затрат от размера проекта (размер выражается в KLOC), вычисляется так же, как и в предыдущих моделях СОСОМО;

в размере проекта различают две составляющие – новый код и N-вариантный (повторно и многократно используемый код);

множитель поправки M_p зависит от 17 факторов затрат, характеризующих продукт, аппаратуру, персонал и проект.

Изменчивость требований при формировании N-вариантных компонент программных систем приводит к повторной работе, требуемой для учета предлагаемых изменений. Оценка их влияния выполняется по формуле

$$K_{req} = l + (BRAK / 100),$$

где $BRAK$ – процент кода, отброшенного (модифицированного) из-за изменения требований.

Размер проекта и продукта определяют по выражению

$$SIZE = SIZE_{new} + SIZE_{reuse} \text{ [KLOC]},$$

где:

$SIZE_{new}$ – размер нового (создаваемого) программного кода;

$SIZE_{reuse}$ – размер повторно используемого программного кода N-вариантных компонент ПО.

Формула для расчета размера повторно используемого кода записывается следующим образом:

$$SIZE_{reuse} = KASLOC \times \left(\frac{100 - AT}{100} \right) \times \frac{AA + SU + 0,4DM + 0,3CM + 0,3IM}{100},$$

где

$KASLOC$ – количество строк повторно используемого кода, который должен быть модифицирован (в тысячах строк);

AT – процент автоматически генерируемого кода;

DM – процент модифицируемых проектных моделей;

CM – процент модифицируемого программного кода;

IM – процент затрат на интеграцию N-вариантных структур ПО, требуемых для подключения многократно/повторно используемого ПО;

SU – фактор, основанный на стоимости понимания добавляемого ПО, изменяется от 50 (для сложного неструктурированного кода) до 10 (для хорошо написанного объектно-ориентированного кода).

Фактор AA отражает стоимость решения о том, может ли он быть повторно используемым в N-вариантных программных структурах мультиверсионной системы. Он зависит от размера требуемого тестирования и поддержки n-вариантного исполнения (величина изменяется от 0 до 8).

Для определения множителя поправки M_p основного уравнения используют 17 факторов затрат, которые могут быть разбиты на 4 категории: факторы продукта; факторы платформы исполнения (виртуальной машины); факторы персонала; факторы проекта. Для каждого фактора определяется оценка (по 6-балльной шкале). На основе оценки для каждого фактора по таблице Боэма определяется множитель затрат EM_j ($j=1, \dots, 17$). Перемножение всех множителей затрат дает множитель поправки постархитектурной модели.

Значение M_p отражает реальные условия выполнения программного проекта и позволяет втрое увеличить (уменьшить) начальную оценку затрат. От оценки затрат легко перейти к стоимости проекта, а после определения затрат и стоимости можно оценить

длительность разработки N-вариантной структуры. Модель COCOMO II содержит уравнение для оценки календарного времени TDEV, требуемого для выполнения проекта. Для моделей всех уровней справедливо:

$$Period(TDEV) = \left[3,0 \times (ЗАТРАТЫ)^{(0,33+0,2(B-1,01))} \right] \times \frac{SCEDPercentage}{100} [мес.],$$

где:

B – ранее рассчитанный показатель степени,

SCEDpercentage – процент увеличения (уменьшения) номинального графика.

Далее в работе рассматриваются четыре модели, каждая из которых применима к различным структурам N-вариантных (или мультиверсионных) систем (от очень простой структуры к более сложной). Многообразие моделей дает проектировщику гибкость в выборе соответствующей модели для проектируемой системы. Рассматривается структура моделей и представлена постановка задачи и методы решения для каждой из них с учетом следующих условных обозначений:

K – число функций N-вариантной программной системы, которые необходимо выполнить;

n – число модулей N-вариантной программной системы;

F_k – частота использования k -й функции ПО, $k = 1, 2, \dots, K$;

m_i – число версий, i -го модуля, $i = 1, \dots, n$;

R_{ij} – оценка надежности j -й версии i -го модуля;

X_{ij} – булева переменная, равная 1, если j -я версия выбрана для i -го модуля, иначе - 0;

R_i – оценка надежности i -го модуля;

R – оценка надежности N-вариантной программной системы;

C_{ij} – стоимость разработки и сопровождения j -й версии i -го модуля;

B – ограничение по стоимости создаваемой N-вариантной программной системы.

Как и ранее, имеют силу основные предположения методологии N-вариантного проектирования ПО, касающиеся, во-первых, использования методов модульного программирования; во-вторых, независимости разработки версий модулей и осуществимости оценки их надежности и стоимости с использованием модулей COTS-сопровождения (эти модули производятся и тестируются независимо, поэтому их надежность может быть оценена с помощью любой из доступных моделей, а их фактическая стоимость – это стоимость закупки и сопровождения); в-третьих, имеется ограничение на стоимость N-вариантной программной системы (N-ВПС).

Обобщающей моделью является модель В-2, формирующая оптимальный состав модулей многофункциональной (K -функций) N-вариантной программной системы с

введением избыточности. Ввиду того, что избыточность в структуре разрешена, возможен выбор более чем одной версии для каждого модуля программной системы. Задача сформулирована следующим образом:

$$\max R = \sum_{k=1}^K F_k \prod_{i \in S_k} R_i,$$

при ограничениях:

$$\begin{aligned} \sum_{j=1}^{m_i} X_{ij} &\geq 1, \quad i = 1, \dots, n; \\ \sum_{i=1}^n \sum_{j=1}^{m_i} X_{ij} C_{ij} &\leq B; \\ X_{ij} &= 0, 1 \quad j = 1, \dots, m_i, \quad i = 1, \dots, n, \\ R_i &= 1 - \prod_{j=1}^{m_i} (1 - R_{ij})^{X_{ij}}. \end{aligned}$$

Надежность i -го модуля определяется как вероятность того, что, по крайней мере, одна из m_i версий выполняется правильно, а совокупность ограничений гарантирует, что для каждого i -го модуля выбрана, по крайней мере, одна версия.

Из-за наличия ограничений и так как мы имеем дело с функциями системы ПО, ни один из методов, предложенных ранее, не может быть «напрямую» использован для решения задачи В-2. Кроме того, целевая функция – нелинейная, поэтому в работе предложен метод линеаризации и представлены численные примеры решения задачи оптимизации состава версий модульных N -вариантных программных структур.

Решение задачи формирования мультиверсионного ПО принадлежит множеству, определяемому следующим выражением

$$S = \bigcup_{i=I_{\min}}^{I_{\max}} O_i(X_0) \tag{1}$$

где X_0 – точка пространства булевых переменных, все координаты которой имеют нулевое значение.

Таким образом, чтобы найти решение задачи, достаточно сравнить значения целевой функции в элементах множества S . Данное свойство позволяет значительно уменьшить вычислительные затраты на поиск точного решения, ускоряя тем самым выполнение процедуры поиска. Мощность множества S определяется следующим выражением

$$\text{card } S = \sum_{k=I_{\min}}^{I_{\max}} C_n^k$$

где C_n^k – число сочетаний из n элементов по k .

Схема поиска на множестве решений, определяемом выражением (1), была реализована и опробована в алгоритме Truncation Exhaustive Search (усеченный полный перебор).

Ограничение на стоимость проектируемого комплекса программ имеет вид

$$C_{NVS1}(X) \leq B$$

$$\text{где } C_{NVS1}(X) = \sum_{i=1}^I \sum_{s=1}^{S_{B_i}} X_s^i \cdot C_{B_i,s} \quad (2)$$

Ограничение на стоимость проектируемого комплекса программ определяется выражением

$$C_{NVS2}(X) \leq B$$

$$\text{где } C_{NVS2}(X) = \sum_{i=1}^I \sum_{k=1}^{J_i} \sum_{s=1}^{J_k^i} X_{ks}^i \cdot C_{j_k^i,s} \quad (3)$$

Для приведенных выше псевдобулевых функций (2), (3) выделяется следующее свойство: если две соседние друг к другу точки X и Y отличаются значением некоторой i -ой компоненты, причем $X_i=0$, а $Y_i=1$, то $C(Y)=C(X)+C_i$, где C_i – стоимость включения в структуру мультиверсии, соответствующей i -й компоненте вектора участия. Полностью значение функции-ограничения для каждой из моделей вычисляется в некоторой начальной точке поиска. Все же последующие значения этой функции получаются прибавлением либо вычитанием соответствующей величины при переходе по соседним точкам.

С целью утилизации вышеописанного свойства построенной оптимизационной задачи была реализована процедура обхода графа булевых переменных по соседним точкам, причем каждая точка области решений просматривается алгоритмом обхода только один раз, что позволяет избежать повышения вычислительной сложности поисковых процедур. Такой способ обхода области поиска был реализован в алгоритме неявного перебора с обходом области поиска по соседним точкам [5]. Поиск осуществляется по направлению от точки $X^0=\{0,\dots,0\}$ к точке $X^1=\{1,\dots,1\}$ – координаты принимают единичные значения. На практике реализация обратного направленного обхода (для этого в блок-схеме нули заменяются единицами, а единицы – нулями) может оказаться более эффективной.

Так, метод изменяющихся вероятностей (МИВЕР) представляет семейство алгоритмов, имеющих общую схему [3]. В применении к задаче формирования мультиверсионного ПО из разработанных ранее алгоритмов схемы МИВЕРа на основе экспериментальных данных наиболее эффективным был признан модифицированный алгоритм случайного поиска с возвратом (МСПА) с независимым распределением вероятностей.

Выше было показано, что вид целевых функций в построенных оптимизационных задачах обуславливает наличие множества постоянства, на котором целевая функция приобретает нулевое значение. Как показала практика, на каждом этапе поиска с использованием алгоритмов СПА (случайный поиск с адаптацией) и МСПА возможна генерация большого числа подобных «нулевых» точек. Поэтому во избежание генерации точек, придающих нулевое значение целевой функции, были разработаны алгоритмы NVS СПА и NVS МСПА, отличающиеся от своих аналогов правилами формирования компонент вектора вероятностей и генерации случайных точек.

Вектор вероятностей «разбивается» на I частей, каждая из которых содержит $L_i, i = \overline{1, I}$ компонент в соответствии с числом программных модулей в формируемом комплексе. Значения компонент вектора вероятностей P определяются таким образом, чтобы сумма значений компонент, соответствующих одному программному модулю, была равна единице, то есть

$$\sum_{l=1}^{L_i} p_l = 1, i = \overline{1, I}$$

При реализации первой оптимизационной модели формирования мультиверсионного программного обеспечения значения параметров L_i определяются как

$$L_{NVS1,i} = S_{B_i},$$

а для второй модели

$$L_{NVS2,i} = S_{j_k}^{j_i}, k = \overline{1, J_i}, i = \overline{1, I}$$

На первом этапе алгоритмов компоненты исходного вектора вероятностей P^0 полагаются равными $1/L_i, i = \overline{1, I}$, что определяет частично независимое распределение вероятностей присвоения единичного значения каждой компоненте $X \in B_2^n$. Случайный выбор конкретного вектора определяется n испытаниями, каждое из которых определяет значение соответствующей компоненты вектора булевых переменных: организуются I циклов испытаний по L_i случайных выборов компонент булевого вектора, которым присваиваются единичные значения.

При тестировании этого алгоритма была выявлена стабильность в нахождении решений. На практике эта особенность алгоритма случайного поиска граничных точек может позволить сократить число вычислений за счет уменьшения параметра L – числа итераций алгоритма.

Предложенные технические и технологические решения и созданные на их основе методы и средства позволяют создать и поддерживать эффективную по затратам времени и ресурсов, независимую от используемой целевой вычислительной платформы и гарантирующую требуемый уровень качества технологию проектирования, разработки и долговременного сопровождения ПО для надёжного доступа к широкополосным мультимедийным услугам.

Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение 14.В37.21.0625.

Список литературы

1. Антамошкин А. Н. Сравнительная эффективность алгоритмов случайного поиска [Текст] / А. Н. Антамошкин. – Томский государственный университет. – Томск, 1979. – 192 с.
2. Ежеманская С. Н., Ступина А. А., Кузьмич Р. И., Вайнгауз А. М., Корпачева Л. Н., Федорова А. В. Многоатрибутивный метод принятия решений на основе качественной информации / С. Н. Ежеманская, А. А. Ступина, Р. И. Кузьмич, А. М. Вайнгауз, Л. Н. Корпачева, А. В. Федорова // Современные проблемы науки и образования. – 2012. – № 5 (43).
3. Ковалев И. В. Мультиверсионное проектирование отказоустойчивого программного обеспечения систем управления / И. В. Ковалев // Научно-технический журнал «Проблемы машиностроения и автоматизации». – Москва, 2006. – № 5. – С. 61-69.
4. Stupina, A. Random point processes [Текст] / A. Stupina, P. Volf // Материалы международной научно-практической конференции САКС. – Красноярск, 2001. – С. 273–276.
5. Stupina, A. The random search algorithms in the space of Boolean variables [Текст] / A. Stupina, A. Antamoshkin // Symp. OR'97. – Jena, 1997. – Pp. 112–118.

Рецензенты:

Петров Михаил Николаевич, д.т.н., профессор, зав. кафедрой электронной техники и телекоммуникаций Сибирского государственного аэрокосмического университета, г.Красноярск.

Пимонов Александр Григорьевич, д.т.н., профессор, профессор кафедры прикладных информационных технологий ФГБОУ ВПО «Кузбасский государственный технический университет имени Т. Ф. Горбачева», г. Кемерово.