

УДК 681.324

Е.И. Бобыр¹, М.А. Скоромнюк², Е.В. Борщ³¹Народная украинская академия, Харьков²Новокаховский политехнический институт, Новая Каховка³Национальный технический университет «ХПИ», Харьков

МЕТОД АНАЛИТИЧЕСКОЙ ОЦЕНКИ ЭФФЕКТИВНОСТИ ПРОЦЕССА РЕАЛИЗАЦИИ КОМПЛЕКСНОГО АЛГОРИТМА ЭВМ АСУ

В статье предложено усовершенствование метода аналитической оценки эффективности реализации функционального программного обеспечения (ФПО) ЭВМ АСУ на ранних стадиях проектирования. Предложенный метод позволяет оценить: эффективность реализации разработанного ФПО, выбрать предпочтительные методы программной защиты в зависимости от уровня требований к надежности вычислений и имеющихся вычислительных ресурсов, оценить допустимую глубину резервирования программ.

Ключевые слова: аналитическая оценка, функциональное программное обеспечение.

Введение

Постановка проблемы. Известно, что конструктивные и схемотехнические методы защиты РЭА не позволяют полностью защитить вычислительные средства АСУ от воздействия помех искусственного и естественного происхождения. Это обстоятельство требует проведения всестороннего анализа возможности и оценки эффективности применения для защиты ЭВМ существующих методов защиты и разработки новых методов, базирующихся на особенностях структуры, организации планирования, контроля и автоматического восстановления вычислительного процесса в перспективных ЭВМ АСУ. К таким новым методам в первую очередь следует отнести программно-алгоритмические методы защиты от искажений кодов программ и оперативной информации, использующие временную, информационную и программную избыточность. При их использовании особую актуальность приобретает проблема оценки эффективности реализации программ операционной системы и функционального программного обеспечения (ФПО) автоматизированных систем управления.

Анализ литературы. Обеспечение надежности реализации программ для ЭВМ АСУ осуществляется применением аппаратно-программных механизмов защиты данных и программных модулей, оперативного контроля хода выполнения программ, дублированием программ и данных на различных носителях, применением системы автоматического восстановления вычислительного процесса после сбоев и отказов в аппаратурных модулях машины (сети ЭВМ) [1 – 4].

К настоящему времени наиболее значительные результаты в области оценки надежности программного обеспечения для ЭВМ АСУ получены в следующих направлениях:

разработаны различные методы оценки надежности аппаратурных средств ЭВМ и программного обеспечения;

разработаны технология структурного программирования и методы тестирования и верификации программ;

разработаны методы повышения надежности программ за счёт избыточности и ограничения взаимного влияния ошибок в различных модулях программ.

Цель данной статьи – усовершенствование метода аналитической оценки эффективности реализации программного обеспечения ЦВС АСУ на ранних стадиях проектирования.

Основная часть

Пусть ФПО системы представлено тройкой:

$$D = \langle A, E, G \rangle,$$

где $A = \{a_i \in A\}_{i=1}^l$ – множество частных алгоритмов (ЧА) или групп частных алгоритмов (ГЧА), входящих в ФПО; E – множество переменных и других входных данных и рабочей информации в D , где каждая переменная e_j принимает значения из области $e_j \cup u$, где u – неопределённость; G – управляющий орграф ФПО:

$$G = \langle W, P \rangle$$

Вершине $w_j \in W$ соответствует оценка вероятности $P(t, t+t_0)$ правильной однократной реализации d_i -го ЧА или ГЧА. Дуге орграфа $p(i, k) \in P$ соответствует оценка вероятности перехода от i -го к k -му ЧА. Эти оценки могут быть получены на основе анализа особенностей реализации ФПО системы (его временной диаграммы и логики выполнения).

Требуется оценить вероятность правильной реализации ФПО $P(t, t+T \nu)$, где $T \nu$ – заданный период времени эксплуатации ФПО.

Оценку вероятности правильной реализации ФПО целесообразно проводить в два этапа. Вначале оценить вероятность её правильной однократной реализации $P(t, t+t_0)$, а затем учесть частоту повторения цикла T_z работы системы.

Будем полагать, что орграф D приведён к структурированному виду, т.е. представим в виде

комбинации (суперпозиции) четырёх основных элементов: конструкции из последовательно выполняемых элементов, конструкции из параллельно выполняемых элементов, конструкции ветвления по условию и циклической конструкции.

Исходим из того, что интенсивность потоков сбоев и отказов в аппаратурных модулях помех несравнима с интенсивностью нарушения процесса реализации ФПО из-за ошибок, оставшихся в программах после отладки, а нарушения управления вычислительным процессом из-за искажений входной и рабочей информации проявляется в неправильной реализации принимаемых решений в системе управления. Ошибочная входная управляющая информация приводит к потере заявки на обработку либо к неправильному выбору подпрограммы, то есть, в конечном итоге к неправильной обработке заявки. Однако будем полагать, что искаженные заявки отбраковываются ещё на входе системы обработки информации.

При введённых допущениях можно сравнительно просто оценить вероятность однократной реализации ФПО $P(t, t+t_0)$ по орграфу D.

Вначале получим выражение для оценки вероятности правильной реализации перечисленных выше элементарных конструкций.

1. Последовательная реализация (рис. 1, а).

Исходя из независимости событий, заключающихся в правильном последовательном выполнении всей совокупности ЧА ФПО, имеем:

$$P_p = \prod_{i=1}^n P_{ai} . \quad (1)$$

2. Параллельное выполнение ЧА (рис. 1, б).

Параллельная реализация характеризуется в общем случае получением m верных решений по n реализуемым ветвям. Обычно $m=n$, т.е. требуется безошибочное выполнение всех n ветвей одновременно.

В этом случае имеем:

$$P_n = \prod_{n=1}^m P_p = \prod_{n=1}^m \prod_{i=1}^n P_{ai} . \quad (2)$$

Однако при реализации методов защищённого программирования параллельное исполнение нескольких вариантов программ должно проверяться на преемственность результатов. При этом получение хотя бы одного приемлемого результата означает окончание реализации параллельной конструкции. Защита программ методом N-версного программирования предполагает совпадение хотя бы двух результатов.

Совместность реализации вариантов означает совместность событий их окончания. Вероятность появления не менее m из n событий равна:

$$P_n^m = \sum_{j=m}^n (-1)^{\uparrow(j-m)} C_{m-1}^{j-1} S_j ; S_j = \sum_{k=j}^n C_{j-1}^{k-1} P_k , m=1, k, \quad (3)$$

где $C(\)$ – сочетание объектов C без повторов.

В соответствии с (3), например, при достаточности хотя бы одного приемлемого результата в двух исполняемых ветвях программ имеем:

$$P_2^1 = P_{a1} + P_{a2} - P_{a1}P_{a2} . \quad (4)$$

3. Ветвление по условию (рис. 1, в). Конструкция “ветвления” – составная конструкция. Её правильное завершение возможно при верном вычислении (с вероятностью P_0) предиката α и приемлемого, с вероятностью P_{ai} , выполнения выбранной с вероятностью $P_i(\alpha)$ ветви a_i . Вероятность приемлемого выполнения каждой ветви оценивается по формуле:

$$P_v = P_0 P_1(\alpha) P_{ai} . \quad (5)$$

Соответственно вероятность приемлемого выполнения всех n ветвей оценивается по формуле:

$$P_v = P_0 \sum_{i=1}^n P_i(\alpha) P_{ai} . \quad (6)$$

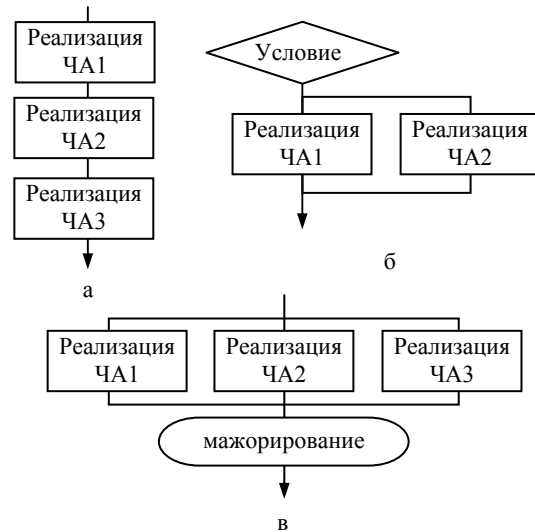


Рис. 1. Типовые конструкции алгоритма
а – последовательная реализация;
б – параллельное выполнение;
в – ветвление по условию

4. Циклическое выполнение (рис. 2).

Среди множества вариантов циклических конструкций, встречающихся в структуре КА, можно выделить две основные: “цикл-до” и “цикл-пока”. К этим двум видам сводятся все остальные варианты повторения участков алгоритма, в том числе и процедурный вызов.

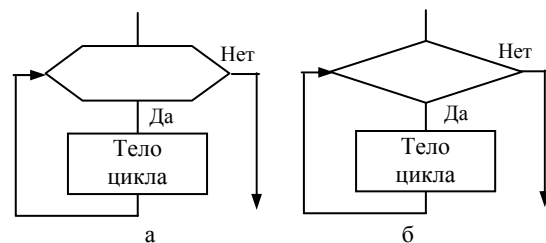


Рис. 2. Типовые конструкции алгоритма при циклическом выполнении: а – “цикл-до”; б – “цикл-пока”

В конструкции „цикл-до” количество повторений заранее известно и равно n . Тем не менее, в ЭВМ, подверженной аппаратурным сбоям и отказам, правильная реализация проверки окончания цикла осуществима не всегда. При описании вероятностными методами обозначим P_0 вероятность правиль-

ного вычисления условия s . Полагая независимость событий приемлемой (безотказной) реализации ЧА a_1 и a_2 и вычисления предиката α , получим:

$$P_{z1} = P_{a1}^{n+1} P_0^{n+1} P_{a2}^n. \quad (7)$$

Рассмотрим конструкцию „цикл-пока”. Цикл заканчивается получением результата за i повторений с вероятностью $P(\alpha, i), i = 1, n$.

Как и ранее, результат вычисления предиката α правилен с вероятностью P_0 . Тогда

$$P_{z2} = \sum_{i=1}^n P_0^i P_1^i P_2^i [1 - P(\alpha, i)] + P_0 P_1 P(\alpha, 0). \quad (8)$$

На основе аналитических выражений (3...8) можно оценить вероятность однократной реализации ФПО $P(t, t+t_0)$. Для получения оценки необходимо, последовательно просматривая оргграф D , заменять участки, соответствующие вышеперечисленным элементарным конструкциям, эквивалентными вершинами с вероятностями

$$P_p, P_v, P_z, P_n^m.$$

Процесс замены продолжается до получения оценок вероятностей правильной (приемлемой) реализации функции ФПО

$$P_{fi}, i = \overline{1, F}.$$

Вероятность правильной однократной реализации ФПО вычисляется в соответствии с выражением:

$$P(t, t+t_0) = \sum_{i=1}^F P_i P_{fi}, \quad (9)$$

где P_i – вероятность появления заявки на i -ю функцию ФПО.

Вероятность правильной реализации ФПО за заданный период $P(t, t+Tv)$ в зависимости от степени готовности программ может определяться различными способами. На начальных стадиях разработки программ, когда оценки носят ориентировочный характер, применима однопараметрическая системно-независимая модель Нельсона [4], в которой параметром служит вероятность безотказной однократной реализации $P(t, t+t_0)$, а аргументом – количество реализаций m :

$$P(t, t+Tv) \cong P(m) \cong 1 - \exp\left[-\sum_{i=1}^m \ln P_i(t, t+t_0)\right]. \quad (10)$$

Верхняя оценка m может быть получена следующим образом: $m = \lceil Tv/t_0 \rceil + 1$, где $\lceil X \rceil$ операция выделения целой части из X .

Выводы

Таким образом, предложенный метод аналитической оценки эффективности реализации программного обеспечения позволяет уже на ранних этапах проектирования функционального ФПО оценить: эффективность реализации разработанного ФПО; выбрать предпочтительные методы программной защиты в зависимости от уровня требований к надежности вычислений и имеющихся вычислительных ресурсов; оценить допустимую глубину резервирования программ.

Эти данные позволяют в дальнейшем при формировании комплекса средств защиты вычислительного процесса в АСУ существенно сократить временные затраты на оценку эффективности средств защиты в составе сети ЦВС.

Список литературы

1. Бобырь Е.И., Лещенко И.Е. Проблема обеспечения надежности вычислений в системах интеллектуального анализа данных на базе сетей ПЭВМ // Сб. науч. тр.: По материалам 1-го Междунар. радиоэлектрон. Форума "Прикладная радиоэлектроника. Состояние и перспективы развития" МРФ – 2002, Харьков, 8-10 окт. 2002 г. – Х., 2002. – Ч. 2. – С. 269-275.
2. Бобырь Е.И., Пантелей Ю.В. Концептуальная модель процесса реализации функциональных алгоритмов ЭВМ АСУ реального времени // Системы обработки информации. – Х.: ХВУ, 1999. – Вып. 1(5). – С. 54-57.
3. Гурко А.И. и др. Программные средства диагностирования ЕС ЭВМ. – М.: Финансы и статистика, 1985. – 213 с.
4. Абчук В.А. и др. Справочник по исследованию операций. – М.: Воениздат, 1979. – 368 с.

Поступила в редколлегию 24.07.2008

Рецензент: д-р физ.-мат. наук, проф. С.В. Смеляков, Харьковский университет Воздушных сил им. И. Кожедуба, Харьков.

МЕТОД АНАЛІТИЧНОЇ ОЦІНКИ ЕФЕКТИВНОСТІ ПРОЦЕСУ РЕАЛІЗАЦІЇ КОМПЛЕКСНОГО АЛГОРИТМУ ЕОМ АСУ

Є.І. Бобир, М.А. Скоромнюк, Є.В. Борщ

У статті запропоновано удосконалення методу аналітичної оцінки ефективності реалізації функціонального програмного забезпечення (ФПО) ЕОМ АСУ на ранніх стадіях проектування. Запропонований метод дозволяє оцінити: ефективність реалізації розробленого ФПО, вибрати переважні методи програмного захисту залежно від рівня вимог до надійності обчислень і наявних обчислювальних ресурсів, оцінити допустиму глибину резервування програм.

Ключові слова: аналітична оцінка, функціональне програмне забезпечення.

METHOD OF ANALYTICAL ESTIMATION OF EFFICIENCY OF PROCESS OF REALIZATION OF COMPLEX ALGORITHM OF COMPUTER AUTOMATIC CONTROL SYSTEM

Ye.I. Bobir, M.A. Skoromnyuk, Ye.V. Borshch

In the article the improvement of method of analytical estimation of efficiency of realization of functional software (FS) COMPUTER is offered to ACE on the early stages of planning. The offered method allows to estimate: efficiency of realization of developed FS, to choose the preferable methods of programmatic defence depending on the level of requirements to reliability of calculations and present calculable resources, to estimate the possible depth of reserving of the programs.

Keywords: analytical estimation, functional software.