УДК 519.714 DOI: 10.47813/mip.4.2022.4.57-63 EDN: KHVCFR

Многоатрибутивный анализ отказоустойчивой программной архитектуры систем мониторинга траектории полета воздушных судов

Д.И. Ковалев[1,2], Т.П. Мансурова[1], Е.В. Туев[3]

[1]Красноярский краевой Дом науки и техники РосСНИО, Красноярск, Россия
[2]Красноярский государственный аграрный университет, Красноярск, Россия
[3]Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, Красноярск, Россия

*E-mail: grimm7jow@gmail.com

Аннотация. В статье рассматривается оригинальная методика декомпозиции отказоустойчивых мультиверсионных программных систем в комплексе мониторинга траектории полета воздушных судов. Используется метод многоатрибутивного анализа, в рамках которого декомпозиция на функциональные подсистемы осуществляется с учетом общности функций, выполняемых программными модулями. В работе атрибут важности функций определяется, исходя из общей функции воздушного судна и множества задач, которые ставятся перед ним. Разделение программных модулей на типы позволяет определить возможные способы повышения их надежности. Рассмотрено мультиверсионное программирование, метод блоков восстановления и схема t/(n-1)-вариантного программирования.

Ключевые слова: многоатрибутивный анализ, система, мониторинг, программное обеспечение, траектория полета

Multi-attribute analysis of fault-tolerant software architecture of aircraft flight path monitoring systems

D.I. Kovalev[1,2], T.P. Mansurova[1], E.V. Tuev[3]

[1]Krasnoyarsk Regional Science and Technology City Hall, Krasnoyarsk, Russia
[2]Krasnoyarsk State Agrarian University, Krasnoyarsk, Russia
[3]Siberian State University of Science and Technology named after academician M.F. Reshetnev, Krasnoyarsk, Russia

*E-mail: grimm7jow@gmail.com

Abstract. The article considers an original method of decomposition of fault-tolerant multiversion software systems in the aircraft flight path monitoring complex. The method of multi-attribute analysis is used, within which decomposition into functional subsystems is carried out taking into account the commonality of functions performed by program modules. In the paper, the attribute of importance of functions is determined based on the overall function of the aircraft and the set of tasks that are set for it. The division of program modules into types makes it possible to determine possible ways to improve their reliability. Multiversion programming, the method of recovery blocks and the scheme of t/(n-1)-variant programming are considered.

Keywords: multi-attribute analysis, system, monitoring, software, flight path

1. Введение

При проектировании сложных программных комплексов, к которым относится система мониторинга траектории полета воздушных судов (ВС) [1-3], возникает задача декомпозиции системы по ряду признаков. В качестве одного из решений в работе рассматривается методика, позволяющая при проектировании выявлять модули и элементы разрабатываемой программной системы, обладающие различными атрибутами [4-6].

Возможными атрибутами могут выступать отдельные элементы, пригодные для резервирования – программные модули в мультиверсионном исполнении [7], а также их функции, типы и явления, возникающие при их отказе.

Разделение программных модулей по выполняемым ими функциям необходимо для обеспечения высоконадежной работы наиболее важных модулей, а, следовательно, и реализуемых функций. Декомпозиция, как процесс разделения, позволяет рассматривать любую исследуемую систему как сложную, состоящую из отдельных взаимосвязанных подсистем, которые, в свою очередь, также могут быть разделены на части. При декомпозиции каждое разделение образует свой уровень [8].

Исходная система располагается на нулевом уровне. После ее разделения получаются подсистемы первого уровня. Разделение этих подсистем или некоторых из них приводит к появлению подсистем второго уровня и т. д.

Функциональная декомпозиция базируется на анализе функций системы. При этом возникает вопрос, что делает система, независимо от того, как она работает. Основанием разбиения на функциональные подсистемы служит общность функций, выполняемых модулями.

2. Постановка задачи

В рамках данной статьи важность функций принимается, исходя из общей функции воздушного судна и множества задач, которые ставятся перед системой мониторинга траектории полета ВС. Причём один и тот же модуль может выполнять несколько функций сразу, а, следовательно, необходимо учитывать, сколько именно функций выполняется, и какая из них является наиболее важной.

При анализе важности компонентов программного обеспечения систем

мониторинга траектории полета ВС следует учитывать, что многие программные модули выполняют системные, а не прикладные задачи, а, значит, важность их должна быть задана выше.

Прикладные же функции могут быть разделены на главные и второстепенные. Такое разделение основывается на том, что прекращение выполнения главной функции приводит к нарушению мониторинга траектории ВС, что в итоге может привести к невозможности вовремя зафиксировать аварийные ситуации или разрушение ВС. Прекращение же второстепенной – к возникновению нарушения в работе системы, т.е. неисправности, не ведущей к аварии.

Программным модулям, выполняющим системные функции, назначается высший приоритет, модулям, участвующим в выполнении главной функции – высокий приоритет, а тем, что выполняют второстепенные – низкий.

Разделение программных модулей на типы позволяет определить возможные способы повышения надежности, применимые к ним.

Возможными типами модулей могут быть модули, пригодные для разных типов проектирования с применением избыточности: мультиверсионное программирование [7-10], блоков восстановления [10-12], t/(n-1) – вариантное программирование [13, 14].

3. Применение t/(n-1) алгоритма в отказоустойчивых системах

Поскольку t/(n-1) алгоритм является одним из малоизвестных алгоритмов, рассмотрим его подробнее. Данный алгоритм предложен Цзе Сюй (Jie Xu) из университета Ньюкастла (University of Newcastle upon Tyne), он основан на t/(n-1) диагностируемости [13]. Часто его называют t/(n-1) алгоритмом принятия решений. Суть алгоритма не в голосовании всех выходов версий, а лишь в сравнении некоторых из них, достаточных для принятия решения.

Рассмотрим систему с количеством версий N=5 и максимальным количеством ошибок t=2. Это будет 2/(5-1) вариант алгоритма. При количестве ошибок, не превышающем t, алгоритм гарантирует выбор правильного варианта из N выходов версий. Однако и при большем количестве неправильных выходов версий система не обязательно выберет ошибочный. С определенной вероятностью произойдет выбор правильного выхода, однако это уже не гарантируется [15].

Рассмотрим алгоритм подробнее на примере 2/(5-1) варианта. Сравниваются попарно выходы четырех версий: 1 и 2, 2 и 3, 3 и 4. Получаем три результата сравнений ω12, ω23, ω34, равные 0, если выходы совпадают и 1, если различаются. На основании только этих трех выходов компараторов алгоритм принимает решение о переключении выхода между выходами 1, 4 и 5 версий. То есть версии 2 и 3 используются только для сравнения, значения их выходов никогда не используются в качестве выхода системы. Схема работы 2/(5-1) алгоритма представлена на рисунке 1.

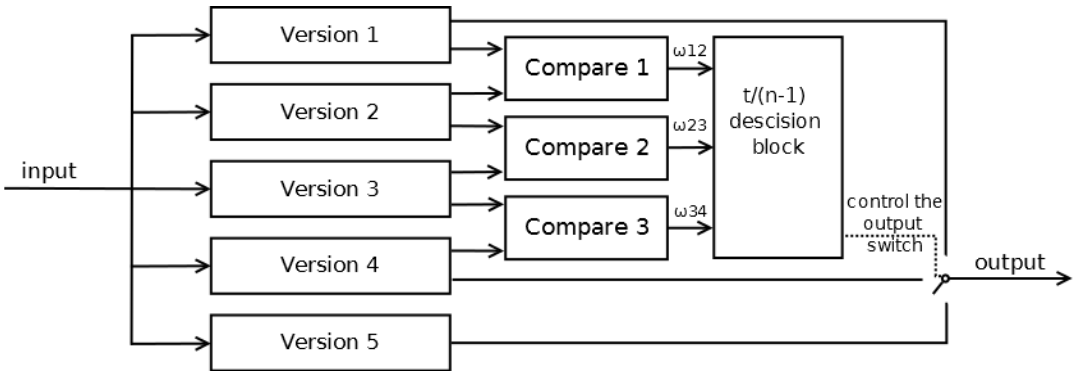

Рисунок 1. Архитектура t/(n-1) алгоритма при n=5 и t=2.

Как видно из рисунка 1, схема принятия решения в t/(n-1) алгоритме относительно не сложна. В случае пяти мультиверсий для принятия решения (правильного управления переключателем выходов) необходимы только результаты трех парных сравнений выходов четырех версий. Значение выхода пятой версий для принятия решения не используется. Логика управления переключателем выходов на основании результатов сравнений при n=5 представлена в таблице 1.

Таблица 1. Возможные варианты выбора на основе выходов компараторов для n=5.

ω_{12}	ω_{23}	ω_{34}	Предположительно правильные версии
0	0	0	1, 2, 3, 4
0	0	1	1, 2, 3
0	1	0	5
0	1	1	1, 2
1	0	0	2, 3, 4
1	0	1	5
1	1	0	3, 4
1	1	1	5

4. Обсуждение результатов

На основании рисунка 1 и таблицы 1, можно сделать вывод, что, при относительно надежных версиях в большей части случаев компараторы будут возвращать (0;0;0) и на выход будет подаваться значение выполнения первой версии.

Так же можно сделать вывод об отсутствии необходимости каждый раз исполнять пятую версию. А делать это только в случае соответствующих значений результатов сравнений, когда на выход необходимо подать именно результат пятой версии ((0;1;0), (1;0;1), (1;1;1)). Этот факт снижает среднюю нагрузку, требуемую среде исполнения мультиверсионного ПО для работы, поскольку в подавляющем большинстве случаев будут рассчитаны 4 из 5 версий. Сам алгоритм принятия решения также является менее ресурсоемким по сравнению с голосованием, особенно с взвешенными его модификациями, где при каждом голосовании исполняются все версии, создаются классы и рассчитываются веса для каждого из них.

Для t/(n-1) при n=5 необходимо только три простые операции сравнения с бинарным выходом. Далее осуществляется однозначный, априорно заданный выбор выхода для одного из восьми возможных сочетаний значений выходов компараторов (Таблица 1).

4. Заключение

Можно сделать вывод, что вычислительная сложность t/(n-1) – алгоритма крайне чувствительна к количеству версий. При добавлении каждой дополнительной версии вычислительная сложность возрастает вдвое. Соответственно при 2 версиях в 4 раза, при 3 дополнительных версиях в 8 раз и т.д. Поскольку главным преимуществом t/(n-1) – алгоритма является именно его низкая ресурсоемкость, наиболее оправдано его применение при количестве версий, не превышающем N=5.

В рамках многоатрибутивного анализа отметим, что под последним рассматриваемым атрибутом – явлением, подразумеваются специфические варианты отказа. Одним из таких специфических отказов может быть опасный отказ, выводящий из строя систему мониторинга траектории полета ВС.

В случае с ВС опасность представляет, в первую очередь, нештатный контакт с поверхностью или другим объектом. Для выявления причин возникновения таких

отказов строятся схемы структуры событий, определяющие, отказы каких модулей самого ВС или системы мониторинга могут привести к возникновению опасного явления. Опасные и безопасные отказы того или иного программного модуля определяются тем, влияет ли этот модуль на передвижение ВС по заданной траектории. Различие опасностей по их влиянию может быть выражено в виде некоторой величины, коэффициента. Следует определить для каждого модуля значение величины *C*, количественно оценивающей опасность, которую несет его отказ. Возможной мерой данной величины, например, для модуля, управляющего направлением движения ВС по траектории может являться скорость движения в данном направлении. То есть, чем выше возможная скорость движения – тем более опасным становится отказ.

Благодарности

Проведение исследований осуществляется при поддержке КГАУ «Красноярский краевой фонд поддержки научной и научно-технической деятельности» в рамках проекта «Контроль траектории полета воздушных судов в экстремальных условиях Арктики и Крайнего Севера» в соответствии с заявкой 2021110907918.

Список литературы

1. Карцан, И. Н. Построение наземных пунктов управления космическими аппаратами с использованием оптимизационно-имитационной модели / И. Н. Карцан // Современные инновации, системы и технологии - Modern Innovations, Systems and Technologies. – 2021. – № 1(2). – С. 64-71. https://doi.org/10.47813/2782-2818-2021-1-2-64-71
2. Серова, Н. А. Транспортная инфраструктура Российской Арктики: специфика функционирования и перспективы развития / Н. А. Серова, В. А. Серова // Проблемы прогнозирования. – 2021. – № 2(185). – С. 142-151.
3. Акзигитов, Р. А. и др. Повышение эффективности мониторинга воздушных судов посредством комплексной системы обнаружения объектов / Р. А. Акзигитов, А. В. Кацура, А. Р. Акзигитов, Д. Е. Строков // «Вестник СибГАУ». – 2016. – № 17(2). – С. 388-392.
4. Ковалев, И. В. К вопросу формирования блочно-модульной структуры системы управления беспилотных летательных объектов / И. В. Ковалев, В. В. Лосев, М. В. Сарамуд, А. О. Калинин, А. С. Лифарь // Современные инновации, системы и технологии - Modern Innovations, Systems and Technologies. – 2021. – №1(3). – С. 48-64. https://doi.org/10.47813/2782-2818-2021-1-3-48-64

5. Зенюткин, Н. В. О способах формирования информационных структур для моделирования объектов, сред и процессов / Н. В. Зенюткин, Д. И. Ковалев, Е. В. Туев, Е. В. Туева // Современные инновации, системы и технологии - Modern Innovations, Systems and Technologies. – 2021. – №1(1). – С. 10-22. https://doi.org/10.47813/2782-2818-2021-1-1-10-22
6. Липаев, В.В. Методы обеспечение качества крупномасштабных программных средств / В. В. Липаев. – М.: Синтег, 2003.
7. Насыров, И. Р. Мультиверсионное программирование в автоматических системах управления технологическими процессами / И. Р. Насыров, Д. В. Сташков // Приоритетные научные направления: от теории к практике. – 2015. – № 16. – С. 75-78.
8. Algirdas, Avizienis. The Methodology of N-Version Programming, in R. Lyu, editor // Software Fault Tolerance. – John Wiley & Sons, 1995.
9. Avizienis, A. The N-Version Approach to Fault -Tolerant Software / A. Avizienis // IEEE Transactions on Software Engineering. – 1985. – № SE-11(12). – С. 1491-1501.
10. Грузенкин, Д. В. Мультиверсонное по и блоки восстановления - два способа защиты от ошибок / Д. В. Грузенкин, О. С. Новиков, А. В. Суханова // Новая наука: от идеи к результату. – 2016. – № 11-2. – С. 72-75.
11. Завьялова, О. И. Модель формирования оптимальной программной системы по схеме блока восстановления с согласованием / О. И. Завьялова, С. Н. Гриценко, С. В. Тынченко, Р. Ю.Царев // Современные проблемы науки и образования. – 2015. – № 1-1. – С. 297.
12. Царев, Р. Ю. Алгоритм формирования программной системы по схеме блока восстановления с согласованием на основе нечеткой логики / Р. Ю. Царев // Современные проблемы науки и образования. – 2015. – № 2. – С. 178.
13. Xu, J. The t(n-1)-diagnosability and its applications to fault tolerance / J. Xu // Digest of Papers. Fault-Tolerant Computing: The Twenty-First International Symposium. – 1991. – С. 496–503.
14. Xu, J. Software Fault Tolerance: t/(n-1)-Variant Programming // Jie Xu, Brian Randell // IEEE Transactions on Reliability. – 1997. – 46(1). – С. 60-68.
15. Kovalev, I. Comparative Tests of Decision Making Algorithms for a Multiversion Execution Environment of the Fault Tolerance Software / I. Kovalev, A. Voroshilova, V. Losev, M. Saramud, M. Chuvashova, A. Medvedev // European Conference on Electrical Engineering and Computer Science (EECS), Bern. – 2017. – С. 211-217.