

УДК: 004.41

DOI: [10.47813/nto.2.2022.5.120-130](https://doi.org/10.47813/nto.2.2022.5.120-130)

EDN: [OHSJIE](https://ojs.urf.edu/edn/OHSJIE)



Анализ транзакционной структуры программного обеспечения в системах мониторинга отказоустойчивых объектов

Т.П. Мансурова¹, Д.И. Ковалев^{1,2}, В.А. Подоплелова^{2,3,*},
А.А. Яблокова¹

¹Красноярский краевой Дом науки и техники РосСНИО, Красноярск, Россия

²Красноярский государственный аграрный университет, Красноярск, Россия

³Сочинский государственный университет, Сочи, Россия

*E-mail: podoplelovava@mail.ru

Аннотация. В статье дается обзор подходов к анализу транзакционной структуры программного обеспечения в системах мониторинга отказоустойчивых объектов. Данные объекты, как правило, являются распределенными (кроссплатформенными) и функционируют в режиме реального времени. Показано, что компьютерные системы мониторинга и контроля объектов, работающих в режиме реального времени, играют очень важную роль в обеспечении их безопасности, поэтому всегда существует потребность в высококачественном программном обеспечении. Целью работы является повышение эффективности анализа транзакционной надежности программных систем обработки информации и мониторинга путем автоматизации моделирования жизненного цикла транзакционной структуры программного обеспечения. Моделирование цикла жизни транзакционной структуры ПО, в первую очередь, связано с формализацией его процессов и этапов, а также с применением методологии, позволяющей описать семантику процессов и этапов формирования, реализации и сопровождения программного обеспечения. Для визуального моделирования цикла жизни транзакционной структуры ПО используется методология IDEF0, позволяющая построить функциональную модель цикла жизни и описания правил данной модели, то есть логические соотношения между работами.

Ключевые слова: мониторинг, отказоустойчивый объект, программное обеспечение, транзакционная структура, надежность

Analysis of the transactional structure of software in monitoring systems for fault-tolerant objects

T.P. Mansurova¹, D.I. Kovalev^{1,2}, V.A. Podoplelova^{2,3,*},
A.A. Yablokova¹

¹Krasnoyarsk Regional Science and Technology City Hall, Krasnoyarsk, Russia

²Krasnoyarsk State Agrarian University, Krasnoyarsk, Russia

³Sochi State University, Sochi, Russia

*E-mail: podoplelovava@mail.ru

Annotation. The article provides an overview of approaches to the analysis of the transactional structure of software in systems for monitoring fault-tolerant objects. These objects, as a rule, are distributed and operate in real time. It is shown that computer systems for monitoring and controlling objects operating in real time play a very important role in ensuring their safety, so there is always a need for high-quality software. The aim of the work is to increase the efficiency of the analysis of the transactional reliability of software systems for information processing and monitoring by automating the simulation of the life cycle of the transactional structure of the software. Modeling the life cycle of a software transactional structure is primarily associated with the formalization of its processes and stages, as well as with the use of a methodology that allows describing the semantics of the processes and stages of the formation, implementation and maintenance of software. For visual modeling of the life cycle of the transactional structure of the software, the IDEF0 methodology is used, which allows you to build a functional model of the life cycle and describe the rules of this model, that is, the logical relationships between works.

Keywords: monitoring, fault-tolerant object, software, transactional structure, reliability

1. Введение

Проектирование современных систем мониторинга отказоустойчивых объектов (космических аппаратов, беспилотных летательных аппаратов, опасных производственных объектов в отраслях машиностроения, энергетики и т.п.) требует надежной обработки информации, что, в свою очередь, предъявляет высокие требования по надежности к программным средствам [1-8]. Очевидно, что компьютерные системы мониторинга и контроля объектов, работающих в режиме реального времени, играют очень важную роль в обеспечении их безопасности, поэтому всегда существует потребность в высококачественном программном обеспечении [9-12].

Надежность программного обеспечения является одним из измеримых аспектов качества программного обеспечения [11-13]. В отличие от аппаратного обеспечения, ПО не стареет, не изнашивается и не ржавеет, поэтому ненадежность программного обеспечения в основном связана с ошибками в коде программ или конструктивными/структурными ошибками в программном обеспечении. Исследователи [13, 14] отмечают, что надежность программного обеспечения является как динамической, так и стохастической. Точное значение надежности программного продукта никогда точно не известно ни в какой момент его жизненного цикла.

Процесс изучения надежности программного обеспечения можно разделить на три этапа: моделирование, измерение и улучшение [14]. Существует множество моделей оценки надёжности ПО, но ни одна из них не может отразить необходимое исследователю количество характеристик программного обеспечения. Не существует единой модели, которая была бы универсальной для всех ситуаций.

Моделирование может имитировать ключевые характеристики процессов, которые связаны с созданием и проверкой документации и кода программ. При этом следует отметить, что измерение надежности программного обеспечения является условным, так как надежность ПО нельзя измерить напрямую. Поэтому для оценки надежности программного обеспечения измеряются другие, связанные с надежностью ПО, факторы. Это, например, надежность транзакций, исполняемых программными средствами в системах мониторинга отказоустойчивых объектов [13].

Повышение надежности программного обеспечения необходимо, однако, труднодостижимо. Однако, ПО можно улучшить за счет достаточного понимания структуры программного обеспечения, характеристик программного обеспечения и средств надежного проектирования программного обеспечения. Полное тестирование

программного обеспечения также невозможно; однако достаточное тестирование и надлежащее сопровождение значительно повысят надежность программного обеспечения. Таким образом, приоритет задачи оценки надежности должен быть выше приоритета задачи ее обеспечения, чего на самом деле не наблюдается [15].

Для изучения программной системы можно экспериментировать с самой системой или с моделью системы, но экспериментировать с самой системой очень дорого и рискованно. Однако цель многих системных исследований состоит в том, чтобы предсказать, как система будет работать, до того, как она будет построена. Следовательно, системные исследования обычно проводятся с моделью системы. Модель — это не только замена системы; это также упрощение системы. Ряд моделей надежности программного обеспечения появился, когда исследователи пытались понять атрибуты ПО, или то, как и почему программное обеспечение выходит из строя, пытались количественно оценить надежность программного обеспечения. С 1970-х годов было предложено более 200 моделей, но вопрос о том, как количественно оценить надежность программного обеспечения, до сих пор остается нерешенным. Не существует единой модели, которую можно было бы использовать во всех ситуациях. Ни одна модель не совершенна; одна модель может хорошо работать для определенного набора входных данных, но совершенно не подходит для решения других задач на другом наборе входных данных [16].

2. Постановка задачи и методы исследования

Большинство существующих аналитических методов для получения показателей надежности программных систем основано на марковских моделях и предположении об экспоненциальном распределении времени отказа. Были предложены методы моделирования роста надежности компонентов, которые не могут быть учтены традиционными аналитическими методами, но они также сталкиваются с проблемой роста масштаба. Имитационная модель, с другой стороны, является привлекательной альтернативой аналитическим моделям, поскольку она описывает систему, характеризуемую ее такими артефактами, как транзакции [17-19], а также связанными с их исполнением событиями, взаимосвязями и взаимодействиями таким образом, что можно проводить эксперименты на модели системы.

В настоящее время транзакционная обработка информации становится одним из важнейших аспектов, определяющим корректность производимых вычислений и

целостность данных, а, следовательно, и характеристики качества программных средств, используемых в системах мониторинга отказоустойчивых объектов [20-24]. Большое количество и постоянно возрастающая сложность программных средств, предусматривающих транзакционную обработку информации, требуют повышенного внимания к моделированию транзакций на протяжении всего жизненного цикла разработки. Реализуемая в программном средстве модель транзакционных вычислений, по сути, определяет, будет ли данное программное средство находиться в целостном состоянии и поддерживать требуемый уровень надежности системы.

Целью данной работы является обзор методов повышения эффективности анализа транзакционной надежности программных систем мониторинга отказоустойчивых объектов путем автоматизации моделирования жизненного цикла транзакционной структуры программного обеспечения.

Для достижения поставленной цели решались следующие задачи:

- анализ транзакционной структуры программного обеспечения систем мониторинга отказоустойчивых объектов (СМОО);
- моделирование жизненного цикла транзакционной структуры программного обеспечения;
- построение модели анализа процесса разработки программного обеспечения;
- синтез универсального представления транзакционной структуры реляционной базы данных системы мониторинга;
- разработка методики анализа транзакционной надежности программного обеспечения СМОО.

При выполнении работы использовались математическое и вероятностное моделирование, теория надежности программного обеспечения, элементы теории вероятностей, объектно-ориентированный анализ, методология IDEF0.

На основе универсального представления транзакционной структуры реляционной базы данных системы мониторинга отказоустойчивых объектов предлагается оригинальная методика анализа транзакционной надежности программных систем, которая позволяет определить целостность состояния и требуемый уровень надежности программного средства на всем жизненном цикле.

Разработанная методика позволяет выполнить моделирование и анализ транзакционной надежности программных систем обработки информации и управления

высокого класса отказоустойчивости, выявить критичные по надежности операционные профили системы, влияющие на надежность функционирования системы в целом.

3. Жизненный цикл транзакционной структуры программного обеспечения

Моделирование цикла жизни транзакционной структуры ПО, в первую очередь, связано с формализацией его процессов и этапов, а также с применением методологии, позволяющей описать семантику процессов и этапов формирования, реализации и сопровождения программного обеспечения.

Для визуального моделирования цикла жизни транзакционной структуры ПО обратимся к методологии IDEF [11], а именно к методологии IDEF0, с целью построения функциональной модели цикла жизни и описания правил данной модели (логические соотношения между работами).

Функциональная модель IDEF0 представляет собой набор блоков, каждый из которых представляет собой «черный ящик» со входами и выходами, управлением и механизмами, которые детализируются (декомпозируются) до необходимого уровня.

В рамках функциональной модели первого уровня в качестве основного блока рассмотрим функцию «Разработка программного обеспечения». Модель интерпретирует определенные задачи, направленные на обеспечение принципа транзакционности. Например:

- «система мониторинга», основанная на операционной системе реального времени FreeRTOS [4];
- выбор аппаратной платформы, на которую портируется FreeRTOS;
- набор подсистем, обеспечивающих основной и «полезный» функционал отказоустойчивого объекта мониторинга [10].

Механизмы управления представлены, в частности:

- техническим заданием, описывающим все аспекты проектирования, тестирования и сопровождения ПО СМОО;
- мультиверсионной методологией, отражающей подходы и принципы реализации представленных функциональных блоков – этапов цикла жизни кроссплатформенного ПО СМОО;
- стандартами, в соответствии с которыми осуществляется долговременное сопровождение ПО, в том числе стандартами эксплуатирующей организации.

В качестве «механизмов» рассматриваются: разработчик ПО, тестировщик и программное обеспечение в комплексе, задействованное на каждом этапе.

Декомпозируя функциональную модель верхнего уровня, получим четыре основных этапа:

1. Разработка архитектурного проекта ПО.
2. Разработка детального проекта ПО.
3. Тестирование компонент ПО.
4. Сопровождение ПО.

Параметры описания функциональной модели верхнего уровня (входные, выходные, управляющие и механизмы) также присутствуют на функциональной модели второго уровня, однако, требуют внимания связующие - выходные и одновременно входные параметры выделенных этапов и их функциональных блоков.

Этап разработки архитектурного проекта ПО [12] связан с описанием системы в терминах составляющих ее модулей (компонентов) и мультиверсий в случае отказоустойчивого исполнения модулей. Стратегия решения требует решения вопросов, касающихся всех архитектурных уровней разрабатываемой системы.

Качество архитектурного проектирования чрезвычайно важно для долговременного успеха системы. Должное архитектурное проектирование позволяет создавать поддерживаемые системы, т.е. системы, поддающиеся пониманию, сопровождению и масштабированию (расширению). Без этих качеств внутренняя сложность программного обеспечения выходит из-под контроля.

Следовательно, крайне важно, чтобы в результате архитектурного проектирования возникла адаптивная система, которая сохранялась бы на этапе программирования и тщательно поддерживалась после поставки системы заказчику, т.е. на этапе сопровождения и эксплуатации. Результатом, и одновременно входным параметром нового этапа, является компонентная структура (архитектура) ПО [16].

На следующем этапе осуществляется функция детального проектирования, т.е. описания внутренней работы каждого компонента программного обеспечения. Его результатом являются подробные алгоритмы и структуры данных для каждого компонента. Поскольку, компоненты разворачиваются на узлах базовой платформы реализации, то соответственно, алгоритмы и структуры данных должны учитывать ограничения (как способствующие, так и препятствующие работе системы), накладываемые на базовую платформу. Результатом, и одновременно входным

параметром нового этапа являются алгоритмы и структуры данных проектируемых компонент, которые участвуют в транзакциях.

4. Заключение

Надежность сложных программных средств, в первую очередь, определяется качеством их компонент - модулей и функциональных групп программ. Высокое качество достигается регламентированным, систематическим тестированием и отладкой. Относительная простота и обозримость большинства компонент позволяет применять достаточно формализованные методы и методики автоматизированного планирования и эффективной реализации тестирования [13]. Применяемые на данном этапе методы и средства обеспечения автоматизации тестирования компонент, в том числе и повторного, существенно сокращают затраты на сопровождение компонент ПО. Это может достигаться как сокращением самого времени тестирования компонент, так и благодаря автоматизации процесса тестирования. Результатом, и одновременно входным параметром нового этапа является сборка компонент под заявленную аппаратную платформу.

Итоговым этапом, рассматриваемым в данном жизненном цикле, является сопровождение ПО. Под долговременным сопровождением понимается управление конфигурацией ПО — это позволяет получить всю транзакционную структуру ПО, т.е. иерархическую связь документов и компонент, представляющих собой конфигурацию проекта; а также управление архивами разработки. Таким образом, результатом этапа и выходным продуктом рассматриваемой модели является документированная транзакционная структура ПО.

References

1. Aljarbough, A. Intellectualization of information processing systems for monitoring complex objects and systems / A. Aljarbough, M. S. Ahmed, M. Vaquera, B. D. Dirting // *Modern Innovations, Systems and Technologies*. – 2022. – № 2(1). – P. 9-17. <https://doi.org/10.47813/2782-2818-2022-2-1-9-17>
2. Kovalev, I. The efficiency analysis of the automated plants / I. Kovalev, P. Zelenkov, S. Ognerubov, K. Bahmareva, E. Denisova // *IOP Conference Series: Materials Science and Engineering*. XVII International Scientific Conference "Reshetnev Readings". – 2015. – 012007. doi: 10.1088/1757-899X/70/1/012007

3. Ковалев, И. В. Оценка надежности асу с блокирующими модулями защиты / Ковалев И. В., Кузнецов П. А., Зеленков П. В., Шайдуров В. В., Бахмарева К. К. // Приборы. – 2013. – № 6(156). – С. 20-23.
4. Ковалев, И. В. К вопросу реализации мультиверсионной среды исполнения бортового программного обеспечения автономных беспилотных объектов средствами операционной системы реального времени / И. В. Ковалев, В. В. Лосев, М. В. Сарамуд, Д. И. Ковалев, М. О. Петросян // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2017. – № 18(1). – С. 58-61.
5. Ковалев, И. В. Оптимизационно-имитационный подход к синтезу автоматизированных систем управления / И. В. Ковалев, Р. Ю. Царев, М. В. Тюпкин, Ю. Д. Цветков // Программные продукты и системы. – 2007. – № 3. – С. 31.
6. Ковалев, И. В. Принятие управленческих решений на основе анализа эффективности организационно-технологических комплексов предприятий / И. В. Ковалев, А. А. Новожилов, Т. А. Рукавицына // Экономика и менеджмент систем управления. – 2011. – № 1(1). – С. 36-42.
7. Ковалев И. В. Мультиверсионный метод повышения программной надежности информационно-телекоммуникационных технологий в корпоративных структурах / И. В. Ковалев, Р. В. Юнусов // Телекоммуникации и информатизация образования. – 2003. – № 2. – С. 50-55.
8. Kovalev, I. The efficiency analysis of automated lines of companies based on DEA method / I. Kovalev, P. Zelenkov, S. Ognerubov // Lecture Notes in Economics and Mathematical Systems. – 2014. – Т. 675. – С. 107-115. doi: 10.1007/978-3-319-03907-7_12
9. Зеленков, П. В. Мультилингвистическая модель распределенной системы на основе тезауруса / П. В. Зеленков, И. В. Ковалев, М. В. Карасева, С. В. Рогов // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2008. – № 1(18). – С. 26-28.
10. Kovalev, I. V. Multiversion environment creation for control algorithm execution by autonomous unmanned objects / I. V. Kovalev, P. V. Zelenkov, V. V. Losev, D. I. Kovalev, N. V. Ivleva, M. V. Saramud // IOP Conference Series: Materials Science and Engineering. 5th International Workshop on Mathematical Models and their

- Applications, 2016, IWMMMA 2016. – 2017. – 173. – 012025. doi: 10.1088/1757-899X/173/1/012025
11. Kovalev, I. V. Model of the reliability analysis of the distributed computer systems with architecture "client-server" / I. V. Kovalev, P. V. Zelenkov, M. V. Karaseva, M. Y. Tsarev, R. Y. Tsarev // IOP Conference Series: Materials Science and Engineering. XVII International Scientific Conference "Reshetnev Readings". – 2015. – 012009. doi: 10.1088/1757-899X/70/1/012009
 12. Ковалев, И. В. Применение СОМ-технологии для реализации мультиверсионного программного обеспечения систем управления и обработки информации, Приборы и системы / И. В. Ковалев, А. А. Ступина, Р. Ю. Царев, В. А. Волков // Управление, контроль, диагностика. – 2007. – № 3. – С. 18-22.
 13. Quyoum, A. Improving Software Reliability using Software Engineering Approach - A Review / A. Quyoum, M. Dar, and S. M. K. Quadri // International Journal of Computer Applications. – 2010. – 10(5). – P. 41-47. <https://www.ijcaonline.org/volume10/number5/pxc3871990.pdf>
 14. Quadri, S. M. K. "Software optimal release policy and reliability growth modeling" / S. M. K. Quadri, N. Ahmad, M. A. Peer // in Proceedings of 2nd National Conference on Computing for Nation Development, INDIACom-2008, (New Delhi, India, 2008). – P. 423-431.
 15. Распопин, Н. А. Модели и методы оптимизации сбора и обработки информации / Н. А. Распопин, М. В. Карасева, П. В. Зеленков, Е. В. Каюков, И. В. Ковалев // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2012. – № 2(42). – С. 69-72.
 16. Ковалев, И. В. Методология оценки и повышения надежности программно-информационных технологий и структур: монография / И. В. Ковалев, Т. И. Семенько, Р. Ю. Царев. – Красноярск, 2005.
 17. Kovalev, I. V. Deriving the optimal structure of n-version software under resource requirements and cost/timing constraints / I. V. Kovalev, K. - E. Grosspietsch // Conference Proceedings of the EUROMICRO. "Proceedings of the 26th Euromicro Conference, EUROMICRO 2000". – 2000. – P. 200-207. doi: 10.1109/EURMIC.2000.874419

18. Ковалев, И. В. К проблеме выбора алгоритма принятия решения в мультиверсионных системах / И. В. Ковалев, А. В. Котенок // Информационные технологии. – 2006. – № 9. – С. 39-44.
19. Ковалев, И. В. Учет субъективных предпочтений ЛПР при мультиверсионном проектировании АСУ / Ковалев И. В., Слободин М. Ю., Царев Р. Ю. Системы управления и информационные технологии. – 2005. – № 1(18). – С. 44-49.
20. Туев, Е. В. Реализация мониторинга эффективности предприятий с помощью специальной подсистемы АСУП / Е. В. Туев, М. Ф. Козлова, О. И. Ольшевская // Современные инновации, системы и технологии - Modern Innovations, Systems and Technologies. – 2021. – 1(2). – С. 34-45. <https://doi.org/10.47813/2782-2818-2021-1-2-34-45>
21. Kovalev, I. V. Analysis of the economic effect of increasing the reliability of information systems of digital agricultural enterprises / I. V. Kovalev, N. A. Testoyedov, E. V. Tuev // IOP Conf. Series: Earth and Environmental Science. – 2020. – № 548 032043. doi:10.1088/1755-1315/548/3/032043
22. Ishankhodjayev, G. Issues of development of intelligent information electric power systems / G. Ishankhodjayev, M. Sultanov, B. Nurmamedov // Modern Innovations, Systems and Technologies. – 2022. – № 2(2). – 0251-0263. <https://doi.org/10.47813/2782-2818-2022-2-2-0251-0263>
23. Kovalev, I. V. Development of machine schedule at engineering enterprises / I. V. Kovalev, M. V. Karaseva, A. A. Voroshilova, E. V. Tuev // IOP Conf. Series: Materials Science and Engineering. – 2019. – № 537 052038. doi:10.1088/1757-899X/537/5/052038
24. Зенюткин, Н. О способах формирования информационных структур для моделирования объектов, сред и процессов / Н. Зенюткин, Д. Ковалев, Е. Туев, Е. 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>
25. Ковалев, Д. И. К вопросу выбора операционной системы реального времени для аппаратно-программной поддержки систем производственно-экологического мониторинга / Д. И. Ковалев, Т. П. Мансурова, Я. А. Тынченко // *Современные инновации, системы и технологии - Modern Innovations, Systems and Technologies.* – 2021. – № 1(2). – С. 46-63. <https://doi.org/10.47813/2782-2818-2021-1-2-46-63>

26. Ковалев, И. В. Анализ эффективности организационно-технологических комплексов предприятий / И. В. Ковалев, А. А. Новожилов, Т. А. Рукавицына // Системы управления и информационные технологии. – 2010. – № 4(42). – С. 36-39.
27. Waseem, M. Blockchain Based Intelligent Transport System / M. Waseem, K. A. Ahmed, M. T. Azeem // Modern Innovations, Systems and Technologies. – 2021. – № 1(3). – P. 70-88. <https://doi.org/10.47813/2782-2818-2021-1-3-70-88>