

References

1. *Tadviser. Gosudarstvo. Biznes. IT* [Tadviser. Government. Business. IT]. Available at: http://www.tadviser.ru/images/3/3d/Cnews_infa_bigdata_4.jpg/. (accessed 15.04.2014).
2. Internet Engineering Task Force. Available at: <http://www.ietf.org/rfc/rfc3164.txt> (accessed 15.04.2014).
3. *Tadviser. Gosudarstvo. Biznes. IT* [Tadviser. Government. Business. IT]. Available at: <http://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:MapReduce>. (accessed 15.04.2014).

© Соболев А. С., 2014

УДК 658.512.001.56

СИСТЕМА ФОРМИРОВАНИЯ СОСТАВА МУЛЬТИВЕРСИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В РЕАЛЬНОМ ВРЕМЕНИ

Е. В. Соловьев

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: blackdeathangel@rambler.ru

Описывается мультиверсионное программное обеспечение и его применение для обеспечения работы системы обработки данных реального времени. Данная методология основывается на введении программной избыточности и позволяет существенно повысить уровень надежности программного обеспечения. В качестве примера реального применения программного обеспечения подобного типа была выбрана система управления системой обработки данных реального времени долговременной орбитальной станции. На основании имеющихся данных о системе управления долговременной орбитальной станции была разработана тестовая задача, представляющая собой план полета из последовательности режимов работы станции с заданными ограничениями. Проведены эксперименты с помощью стандартного алгоритма муравьиной колонии на разработанной тестовой задаче.

Ключевые слова: оптимизация, муравьиные алгоритмы, мультиверсионное программное обеспечение, системы реального времени.

THE SYSTEM OF FORMATION OF THE COMPOSITION MULTIVERSIONED VIEWS OF THE SOFTWARE IN REAL TIME

Y. V. Solovyev

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochoy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: blackdeathangel@rambler.ru

The article describes software multiversioned views and its application to ensure that the system is processing real-time data. This methodology is based on the introduction of software redundancy and allows to significantly increase the reliability of the software. As a practical example of application software of this type management system, the data processing system of real-time long-term orbital station was chosen. Based on the available data about the long-term orbital station to the task of representing the flight plan from a sequence of modes of operation of the station, with specified restrictions was developed. The experiments were conducted using standard ant colony algorithm developed on the test task.

Keywords: optimization, ant algorithms, multiversioned views software, real time systems.

В современном мире огромное значение приобрела вычислительная техника и в особенности программное обеспечение (ПО), осуществляющее на ней свою работу. В настоящее время благодаря развитию технологий и улучшению аппаратных средств влияние программного обеспечения на надежность и стоимость вычислительных систем значительно возросло. Любое современное программное обеспечение можно считать сложной системой. Существует особый класс ПО, критичный ко времени выполнения – системы реального времени. Данный класс ПО является одним из самых сложных и важных типов ПО, так как чаще всего данное ПО задействовано в управлении объектами, выход из строя которых может привести к значительным экономическим потерям, физическим повреждениям или создавать угрозу человеческой жизни. Поэтому большое значение имеют механизмы, способные повысить надежность подобных систем.

Для систем, критических по надежности, используются специальные методы программирования, гарантирующие безопасность, защищенность и безотказность программных систем. Один из подходов для создания высоконадежного ПО, устойчивого к сбоям, заключается в использовании отказоустойчивых архитектур. Отказоустойчивые архитектуры – это архитектуры аппаратных и программных средств, обеспечивающие устойчивость к сбоям. Такие архитектуры включают резервирование аппаратных и программных средств и имеют блок анализа сбоев, который обнаруживает и устраняет ошибки. Одной из таких архитектур является мультиверсионное программирование [1].

При использовании мультиверсионного программирования в соответствии с общей спецификацией различными командами разработчиков разрабатывается несколько версий программного обеспечения. Эти версии осуществляют вычисления, а результаты их работы сравниваются с помощью системы согласования. Результаты версии, не совпадающие с половиной версий и еще одной или не полученные вовремя, отвергаются [2; 3].

Одним из примеров, где ПО подобного типа может использоваться, является система управления обработки данных реального времени долговременной орбитальной станции (ДОС). Под долговременными орбитальными станциями понимают специально созданные и соответствующим образом оборудованные сооружения и комплексы сооружений, расположенные в открытом космосе или на каких-либо космических телах и предназначенные для научной, производственной, испытательной, эксплуатационной или иной деятельности людей и их жизни в течение длительного промежутка времени. Современная ДОС представляет собой комплекс технических средств, функционально связанных между собой в процессе выполнения поставленных перед ними задач.

Управление работой станции, обеспечение условий жизнедеятельности ее экипажа возложены на бортовые вычислительные машины и системы, обес-

печивающие функционирование различных систем обработки данных и управление в реальном масштабе времени, мониторинг изменений, а также перестроение состава задействованных модулей ПО при использовании мультиверсионного программирования. Основной проблемой является управление всеми составляющими системы для обеспечения максимальных показателей эффективности с учетом поставленных ограничений.

В качестве основного средства управления существуют режимы станции и план ее полета, который представляет собой упорядоченный набор режимов станции. Под режимом ДОС понимают множество выполняемых функций, необходимых для реализации конкретного режима. Ему соответствует определенный набор программных средств. В конкретный момент времени выполняется только один режим [4].

В качестве тестовой задачи был выбран следующий план полета:

1. Стандартный режим: $0 \leq t < 100$, $C \rightarrow \min$, $R_{\min} = 0,993$.
2. Стыковка: $100 \leq t < 200$, $R \rightarrow \max$, $C_{\max} \rightarrow 1600$.
3. Стандартный: $200 \leq t < 300$, $C \rightarrow \min$, $R_{\min} = 0,995$.
4. Внешняя деятельность: $300 \leq t < 400$, $R \rightarrow \max$, $C_{\max} \rightarrow 1300$.
5. Стыковка: $400 \leq t < 500$, $R \rightarrow \max$, $C_{\max} \rightarrow 1600$.
6. Коррекция орбиты: $500 \leq t < 600$, $R \rightarrow \max$, $C_{\max} \rightarrow 1500$.
7. Микрогравитация: $600 \leq t < 700$, $C \rightarrow \min$, $R_{\min} = 0,991$.
8. Стандартный: $700 \leq t < 800$, $C \rightarrow \min$, $R_{\min} = 0,995$.

Для проверки возможности эффективной работы подобного ПО был создан программный комплекс, получивший название АСО N-version software creator. Данное программное приложение было написано на языке программирования C# с использованием среды программирования приложений Microsoft Visual Studio 2012. Это позволило использовать возможности, предоставляемые объектно-ориентированным подходом при разработке программного обеспечения. Схема разработанного комплекса представлена на рис. 1.

Основное функционирование данного ПО обеспечивает модуль Program. Модуль осуществляет ряд шагов для начала процесса работы мультиверсионного программного обеспечения с динамической архитектурой, работающего в реальном времени:

1. Запуск системного времени, что позволяет нам отслеживать время работы программы, а также изменений, которые возникают в процессе ее работы.
2. Создание и запуск объекта, который осуществляет мониторинг изменений и применение изменений, возникающих в процессе работы программы.
3. Остановку поиска по завершению процесса работы программы, что включает в себя остановку монитора, следящего за возникающими изменениями.

Для обеспечения процесса работы мультиверсионного программного обеспечения с динамической

архитектурой используются три взаимосвязанных объекта: монитор изменений, симулятор работы программы, а также конструктор решений. Монитор изменений является объектом, который через определенные промежутки времени проверяет наличие изменений в программе, также он отвечает за непосредственный запуск процесса построения решения. Конструктор решений является объектом, который осуществляет взаимодействие с объектом, ищущим оптимальный состав мультиверсионного программного обеспечения, а также осуществляет хранение лучшего найденного решения. Симулятор работы программы является объектом, который осуществляет симуляцию работы программы, сохраняя параметры найденного решения через определенные кванты времени для их последующего анализа, непосредственно осуществляет реальную работу, возложенную на ПО.

Благодаря наличию этих трех элементов ПО подобного типа в полной мере способно осуществлять свою работу. Основной вопрос, который встает перед разработчиком, является выбор алгоритма для конструирования состава мультиверсионного программного обеспечения. В данном случае выбор был сделан в пользу алгоритма муравьиной колонии, так как динамическая среда исполнения предполагает изменения в структуре и составе мультиверсионного ПО [5; 6].

В результате работы данной программы на разработанной тестовой задаче было установлено, что данное ПО способно обеспечивать формирование состава мультиверсионного ПО с динамической архитектурой с заданными ограничениями. На рис. 2 и 3 представлены графики изменения надежности и стоимости найденного решения в процессе работы алгоритма.

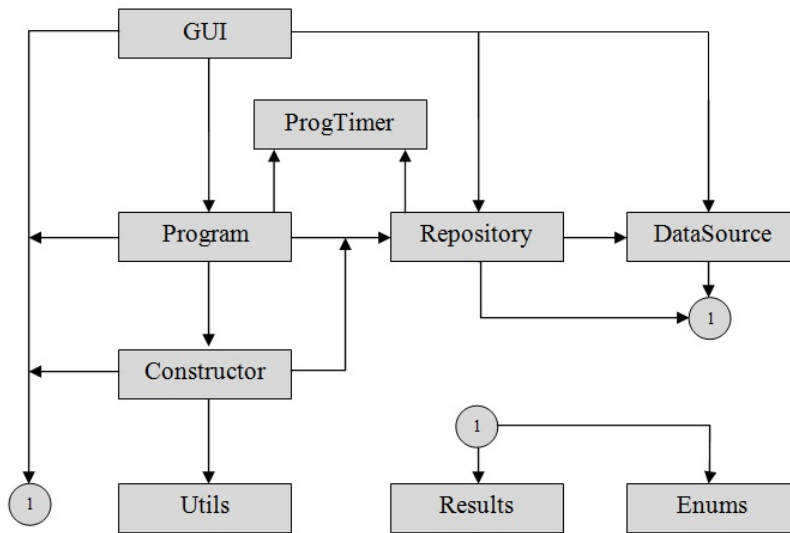


Рис. 1. Схема разработанного программного комплекса

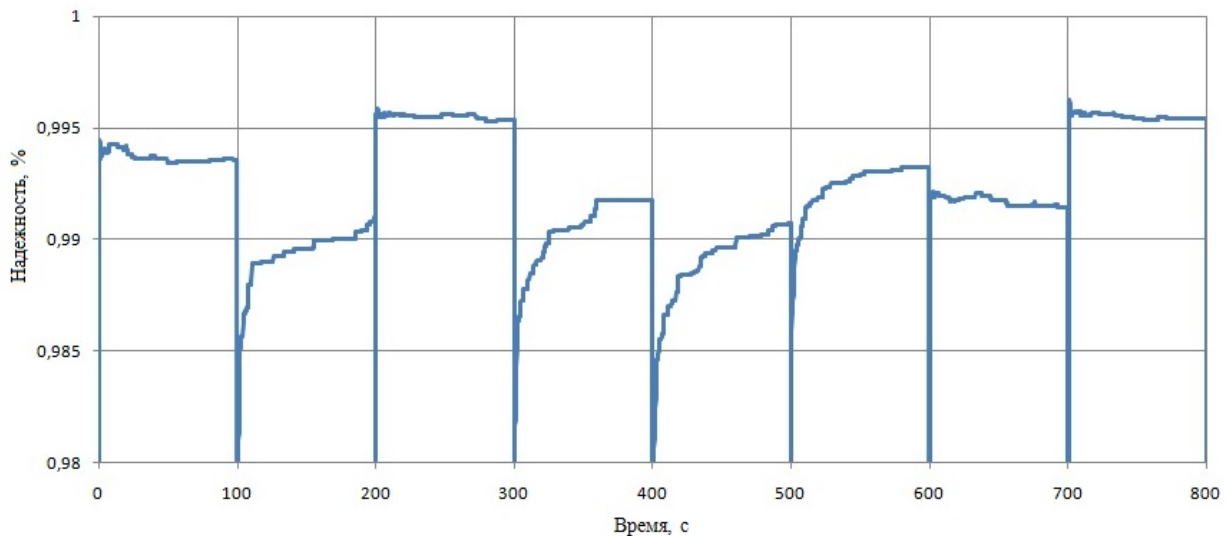


Рис. 2. График изменения надежности найденного решения

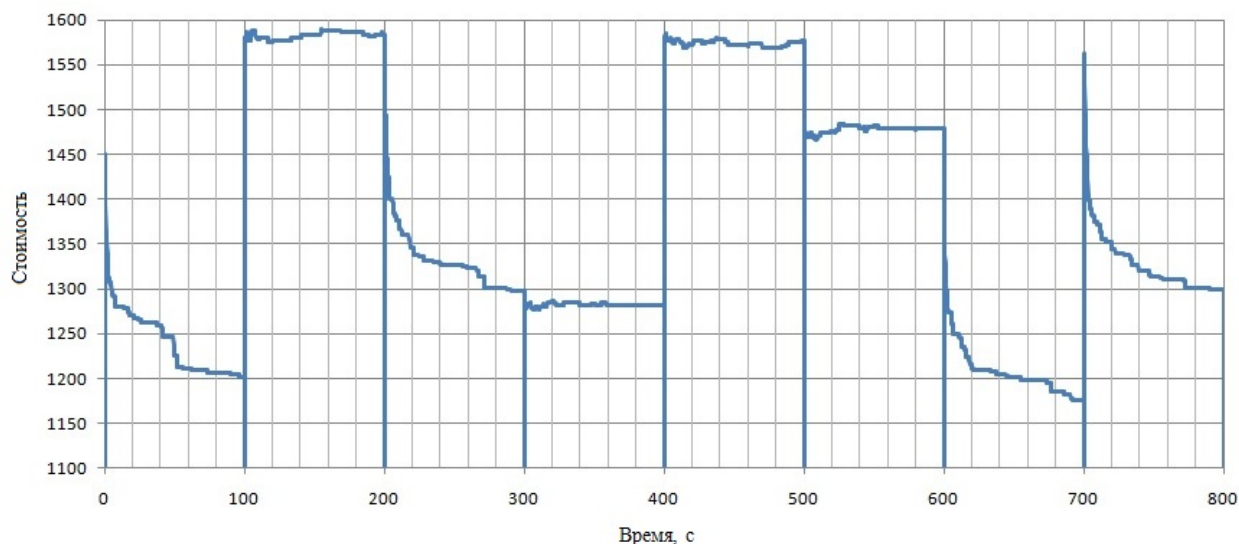


Рис. 3. График изменения стоимости найденного решения

Разработанная система формирования состава мультиверсионного программного обеспечения на практике доказала свою эффективность. На разработанной тестовой задаче, в основу которой легли данные о системе обработки данных реального времени долговременной орбитальной станции, система демонстрирует способность нахождения состава мультиверсионного ПО за ограниченное время и его последующего улучшения в процессе работы алгоритма муравьиной колонии, который был выбран в качестве основного алгоритма для нахождения состава мультиверсионного программного обеспечения.

Библиографические ссылки

1. Соммервиль И. Инженерия программного обеспечения, 6-е изд. М. : Вильямс, 2002. 624 с.
2. Bishop P. G. Review of Software Design Diversity // Сайт компании Adelard. URL: <http://www.adelard.com/papers/divchap.pdf> (дата обращения 26.05.2014).
3. Ковалев И. В., Слободин М. Ю., Ступина А. А. Математическая постановка задачи проектирования N-версионных программных систем // Проблемы машиностроения и автоматизации. 2005. № 3. С. 16–23.
4. Кульба В. В., Микрин Е. А., Павлов Б. В. Проектирование информационно-управляющих систем долговременных орбитальных станций. М. : Наука, 2002. 343 с.
5. Dorigo M., Stützle Th. Ant colony optimization. Cambridge: The MIT Press, 2004.

6. Binitha S., Siva Sathya S. A Survey of Bio inspired Optimization Algorithms // International Journal of Soft Computing and Engineering. 2012. Vol. 2, Iss. 2. P. 137–151.

References

1. Sommervall' I. *Inzheneriya programmnogo obespecheniya* [Software Engineering]. Moscow, Vil'jams Publ., 2002, 624 p.
2. Bishop P. G., Review of Software Design Diversity. Available at: <http://www.adelard.com/papers/divchap.pdf> (accessed 26.05.2014).
3. Kovalev I. V., Slobodin M. Ju., Stupina A. A. [The mathematical formulation of designing N- versioned software systems]. *Problemy mashinostroeniya i avtomatizacii*. 2005, no. 3, p. 16–23. (In Russ.)
4. Kul'ba V. V., Mikrin E. A., Pavlov B. V. *Proektirovanie Informatsionno-upravljajushhikh sistem dolgovremennykh orbital'nykh stantsij* [Design information management systems of long-term orbital stations]. Moscow, Nauka Publ., 2002, 343 p.
5. Dorigo M., Stützle T. Ant colony optimization. Cambridge, Massachusetts, The MIT Press, 2004.
6. Binitha S., Siva Sathya S. A Survey of Bio inspired Optimization Algorithms. *International Journal of Soft Computing and Engineering*, 2012, Vol. 2, Iss. 2, p. 137–151.

© Соловьев Е. В., 2014