УДК 681.34

УЧЕТ СОВМЕСТИМОСТИ COTS-КОМПОНЕНТОВ ПРИ ФОРМИРОВАНИИ ИЗБЫТОЧНЫХ ПРОГРАММНЫХ СИСТЕМ

Царев Р. Ю., Завьялова О. И., Черниговский А. С.

ФГАОУ ВПО «Сибирский федеральный университет», Красноярск, Россия (660074, Красноярск, ул. Академика Киренского, 26, УЛК), e-mail: tsarev.sfu@mail.ru

Создание крупных программных систем вынуждает использовать программные компоненты не только собственной разработки, но и готовые программные компоненты, представленные сторонними разработчиками и поставщиками на рынке программных продуктов. Особенно актуально их использование для программных систем, к которым особые требования по надежности вынуждают прибегать к введению программной избыточности как средству повышения уровня надежности. Применение готовых программных компонентов, именуемых на международном рынке *commercial off-the-shelf* или *COTS*-компонентами, в составе программной системы требует учета совместимости компонентов. В данной статье представлена модель оптимизации, предназначенная для формирования надежной программной системы с учетом совместимости *COTS*-компонентов. Программная система реализуется согласно схеме блоков восстановления с согласованием, что позволяет обеспечить высокий уровень надежности.

Ключевые слова: надежность, избыточность, оптимизация, COTS.

AN ACCOUNT OF COTS COMPONENTS COMPATIBILITY WHEN DESIGN A REDUNDANT SOFTWARE SYSTEM

Tsarev R. Y., Zavyalova O. I., Chernigovskiy A. S.

Siberian Federal University, Krasnoyarsk, Russia (660074, Krasnoyarsk, street A. Kirenskogo, 26, ULK), e-mail: tsarev.sfu@mail.ru

Construction of large software systems is forcing to use software components both of home design and provided by third-party developers and suppliers at the market of software products. The use of the latter is especially important for the software systems with high reliability requirements where the introduction of software redundancy is chosen as a tool for reliability improving. The application of so-called commercial off-the-shelf or COTS components as part of a software system requires consideration of components compatibility. The article presents an optimization model for synthesis a highly reliable software system based on consensus recovery block scheme. The optimization model takes into account of COTS components compatibility.

Keywords: reliability, redundancy, optimization, COTS.

В связи с развитием информационных технологий и применением вычислительной техники в промышленной и финансовой сфере крайне важным становится обеспечение высокого уровня надежности, поскольку невыполнение данных требований может привести к существенным убыткам. Повышенные требования по надежности предъявляются и к программному обеспечению вычислительных машин и систем [1].

Одним из актуальных на сегодняшний день подходов к созданию надежного программного обеспечения является применение блоков восстановления [10]. Программное обеспечение, разрабатываемое согласно данному подходу, отличается применением избыточных программных компонент, предназначенных для решения одной и той же задачи. В контексте программного обеспечения схемы блоков восстановления данные программные компоненты называются альтернативами [6, 10].

Фактически блоки восстановления комбинируют основные идеи контрольных точек и рестарта для компонент программного обеспечения таким образом, что различные альтернативы используются только после того, как обнаруживается ошибка.

Контрольные точки определяются во время проектирования программного обеспечения и служат для того, чтобы восстановить состояние ПО, если в альтернативе произойдет сбой или она вернет неверный результат. Оценка результата производится посредством приемочного теста. В том случае, если результат работы альтернативы оказался неверным, происходит откат и выполнение другой альтернативы [8]. Следует отметить, что приемочный тест должен обладать некоторой априорной информацией о предполагаемых результатах, которая позволила бы оценить полученные результаты работы альтернатив.

Естественным развитием подхода с применением блоков восстановления стало его объединение с альтернативным подходом, где функционально эквивалентные программные компоненты исполняются параллельно и результаты их работы оцениваются посредством алгоритмов голосования [4]. Последний получил название мультиверсионного программирования, а программные компоненты – мультиверсии или просто версии [3]. Синтез этих двух подходов привел к появлению блоков восстановления с согласованием, где каждая альтернатива представлена набором версий, т. е. каждая альтернатива является мультиверсионной [6].

Концепция программной избыточности предполагает формирование структуры программного обеспечения, используя набор программных компонент. Это справедливо как для мультиверсионного программного обеспечения, как и программного обеспечения с блоком восстановления, в том числе, с блоком восстановления с согласованием. Возможность модульной декомпозиции каждого из этих классов программного обеспечения позволяет использовать программные компоненты не только собственной разработки, но и созданные сторонними разработчиками и соответствующие требуемой спецификации.

На международном рынке программных продуктов это программные компоненты получили название *COTS*-компонентов (от англ. *commercial off-the-shelf* – «коробочный программный продукт») [9].

Реализация в программных компонентах различных методов и алгоритмов, предназначенных для решения одной и той же задачи, позволяет избежать возникновения идентичных ошибок в этих компонентах и тем самым обеспечить отказоустойчивое исполнение всего программного обеспечения. Компоненты могут быть представлены в виде библиотек, часть из них может быть разработана самостоятельно, а часть – различными поставщиками. Однако в некоторых случаях программные компоненты могут быть

несовместимы из-за проблем интерфейса, спецификации или реализации [5]. В связи с этим возникает необходимость учета совместимости различных программных компонент.

В данной статье предложена модель оптимизации, предназначенная для решения задачи формировании высоконадежного программного обеспечения с учетом совместимости входящих в его состав программных компонент, в первую очередь, *COTS*-компонентов. Предлагаемая модель является развитием модели оптимизации, предложенной авторами в [2].

Показатели надежности и стоимости программных компонент

Будем рассматривать программную систему, реализуемую согласно схеме блоков восстановления с согласованием. Положим, что данная программная система состоит из n модулей и выполняет L функций. При этом для выполнения функций используются различные (пересекающиеся) множества модулей системы. Обозначим частоту выполнения l-й функции как f_l, а количество модулей, необходимых для ее выполнения, как s_l, $s_l \in S$, где S представляет собой архитектуру программной системы. Поскольку программная система создается согласно схеме блоков восстановления с согласованием, то для i-го модуля, $i \in 1, ..., n$, доступны m_i альтернатив, тогда для j-й альтернативы i-го модуля, $i \in 1, ..., n$, $j \in 1, ..., m_i$, доступны V_{ij} версий.

Задачу оптимизации можно описать следующим образом: необходимо выбрать такой набор версий альтернатив программной системы с блоком восстановления с согласованием, который обеспечил бы максимальные уровень надежности при заданных ограничениях. Поскольку в данный набор входят программные компоненты, созданные сторонними разработчиками – *COTS*-компоненты, требуется учесть их совместимость и возможность интеграции в состав программной системы.

Перед тем как перейти к формальной постановке задачи, рассмотрим показатели надежности, а также особенности избыточной реализации программной системы с блоком восстановления с согласованием.

Отдельная альтернатива может быть, либо приобретена (*COTS*-компонент), либо разработана самостоятельно. Обозначим эту взаимосвязь следующим образом:

$$y_{ij} + \sum_{k=1}^{V_{ij}} x_{ijk} = 1;\ i = 1, 2, ..., n;\ j = 1, 2, ..., m_i,$$

где x_{ijk} – булева переменная, равная 1, если выбранная k-я версия j-й альтернативы i-го модуля, создана сторонними разработчиками, 0 – если данная версия разработана самостоятельно; y_{ij} – булева переменная, равная 1, если j-я альтернатива i-го модуля разрабатывается собственными силами, 0 – в противном случае.

При этом обеспечивается избыточность компонент программной системы на основе *COTS*-компонентов и компонентов собственной разработки:

$$y_{ij} + \sum_{k=1}^{V_{ij}} x_{ijk} = z_{ij},$$

$$\sum_{j=1}^{m_i} z_{ij} \geq 1;\ i = 1, 2, ..., n,$$

где z_{ij} – булева переменная, равная 1, если j-я альтернатива присутствует в i-м модуле, 0 – в противном случае.

Определение вероятности отказа j-й альтернативы i-го модуля, разработанной собственными силами, может быть выполнено на основе числа успешных тестов [7]:

$$N_{ij}^{+} = (1 - q_{ij}) N_{ij};\ i = 1, 2, ..., n;\ \ j = 1, 2, ..., m_i,$$

где N_{ij}^{+} – количество успешно пройденных тестов; N_{ij} – общее количество тестовых испытаний; q_{ij} – вероятность того, что единичное тестирование j-й альтернативы i-го модуля будет неуспешным.

Тогда вероятность безотказной работы компонента собственной разработки можно определить как:

$$p_{ij} = \frac{1 - q_{ij}}{(1 - q_{ij}) + q_{ij}(1 - q_{ij}) N_{ij}^{+}};\ i = 1, 2, ..., n;\ \ j = 1, 2, ..., m_i.$$

В том случае, если используется *COTS*-компонент, т. е. программный компонент, созданный сторонними разработчиками, то значения его характеристик, в том числе, показателей надежности предоставляется поставщиком.

Надежность j-й альтернативы i-го модуля программной системы можно представить в виде:

$$R_{ij} = p_{ij} y_{ij} + \sum_{k=1}^{V_{ij}} R_{ijk} x_{ijk};\ i = 1, 2, ..., n;\ \ j = 1, 2, ..., m_i,$$

где R_{ijk} – вероятность безотказной работы k-й версии j-й альтернативы i-го модуля (вероятность безотказной работы *COTS*-компонента).

Стоимость разработки j-й альтернативы i-го модуля равна величине $c_{ij}(t_{ij} + \tau_{ij} N_{ij})$, где c_{ij} – стоимость единичной разработки j-й альтернативы i-го модуля, t_{ij} – расчетное время разработки j-й альтернативы i-го модуля, τ_{ij} – среднее время, требуемое на выполнение тестового сценария для j-й альтернативы i-го модуля.

При выборе покупке готовых *COTS*-компонентов срок поставки k-й версии j-й альтернативы i-го модуля, обозначаемый как d_{ijk}, указывается поставщиком. Время же

собственной разработки компонентов рассчитывается разработчиками программной системы по формуле $t_{ij} + \tau_{ij}N_{ij}$.

Для создания программной системы требуются временные затраты, связанные с интеграцией различных компонентов в состав системы, тестированием, а также поставкой *COTS*-компонентов. Ограничение на весь срок разработки программной системы может быть определен следующим образом:

$$y_{ij}(t_{ij} + \tau_{ij}N_{ij}) + \sum_{k=1}^{V_{ij}} d_{ijk}x_{ijk} \le T;\ i = 1, 2, ..., n;\ j = 1, 2, ..., m_i.$$

где T - максимальный срок разработки программной системы.

Модель формирования программной системы с блоком восстановления с согласованием с учетом совместимости COTS-компонентов

Модель формирования оптимальной программной системы по схеме блока восстановления с согласованием, предложенная авторами в [2], предполагала, что альтернативы, представленные *COTS*-компонентами, одного модуля совместимы с альтернативами, также являющиеся *COTS*-компонентами, других модулей. Однако в некоторых случаях альтернативы модулей могут быть несовместимы с альтернативами других модулей [5].

Для решения данной проблемы необходимо учесть ряд дополнительных ограничений. В частности, одно из ограничений может быть представлено в следующем виде:

$$x_{abe} \le x_{vu_t w},$$

которое означает, что если для модуля a выбрана альтернатива b, тогда для модуля v должна быть выбрана альтернатива u_t, $t = 1, \ldots, z$. При этом считается, что, если две альтернативы совместимы, то их версии тоже совместимы.

Запишем ограничения, связанные с совместимостью различных альтернатив модулей:

$$x_{abe} - x_{vu_t w} \le My_t;\ \ e = 2, ..., V_{ab};\ \ w = 2, ..., V_{vu_t};\ \ b = 1, ..., m_a,$$

$$\sum y_t = z(V_{vu_t} - 2),$$

где y_t – булева переменная, равная 1, если выбрана одна пара из разных пар альтернатив модулей, 0 – в противном случае.

Можно отметить, что, если необходимо подобрать более одного компонента совместимого с текущей альтернативой, то последнее ограничение может быть ослаблено:

$$\sum y_t \le z(V_{vu_t} - 2).$$

Модель формирования программной системы с учетом совместимости *COTS*-компонентов, призванная максимально увеличить надежность системы с одновременной минимизацией ее общей стоимости, может быть записана следующим образом:

$$\max R = \sum_{l=1}^{L} f_l \prod_{i \in s_l} R_i, \ \min C = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(c_{ij} (t_{ij} + \tau_{ij} N_{ij}) y_{ij} + \sum_{k=1}^{V_{ij}} c_{ijk} x_{ijk} \right)$$

при условиях:

$$R_i = 1 + \left[\sum_{j=1}^{m_i} \frac{1}{(1-R_{ij})^{z_{ij}}} \left(\prod_{k=1}^{m_i} (1-R_{ik})^{z_{ik}} \right) (1-(1-R_{ij})^{z_{ij}}) + \prod_{j=1}^{m_i} (1-R_{ij})^{z_{ij}} \right] \cdot$$

$$\cdot \left[\sum_{j=1}^{m_i} z_{ij} \left(\prod_{k=1}^{j-1} P(A_{ik})^{z_{ik}} \right) P(B_{ij}) - 1 \right]; \ i = 1, 2, ..., n,$$

$$P(A_{ij}) = (1-p_1)[(1-R_{ij})(1-p_3) + R_{ij} p_2],$$

$$P(B_{ij}) = R_{ij}(1-p_2),$$

$$R_{ij} = p_{ij} y_{ij} + \sum_{k=1}^{V_{ij}} R_{ijk} x_{ijk}; \ i = 1, 2, ..., n \text{ и } j = 1, 2, ..., m_i,$$

$$p_{ij} = \frac{1-q_{ij}}{(1-q_{ij}) + q_{ij}(1-q_{ij}) N_{ij}^{+}}; \ i = 1, 2, ..., n \text{ и } j = 1, 2, ..., m_i,$$

$$N_{ij}^{+} = (1-q_{ij}) N_{ij}; \ i = 1, 2, ..., n \text{ и } j = 1, 2, ..., m_i,$$

$$y_{ij} + \sum_{k=1}^{V_{ij}} x_{ijk} = 1; \ i = 1, 2, ..., n; \ j = 1, 2, ..., m_i,$$

$$y_{ij} + \sum_{k=1}^{V_{ij}} x_{ijk} = z_{ij}, \ i = 1, 2, ..., n; \ j = 1, 2, ..., m_i,$$

$$\sum_{j=1}^{m_i} z_{ij} \geq 1; \ i = 1, 2, ..., n,$$

$$y_{ij} (t_{ij} + \tau_{ij} N_{ij}) + \sum_{k=1}^{V_{ij}} d_{ijk} x_{ijk} \leq T; \ i = 1, 2, ..., n; \ j = 1, 2, ..., m_i.$$

$$x_{abe} - x_{vu_t w} \leq M y_t; \ e = 2, ..., V_{ab}; \ w = 2, ..., V_{vu_t}; \ b = 1, ..., m_a,$$

$$\sum y_t = z(V_{vu_t} - 2),$$

где R – надежность (вероятность безотказной работы) программной системы; R_i – надежность i-го модуля; R_{ij} – надежность j-й альтернативы i-го модуля i; C – общие расходы на разработку и закупку компонент программной системы; c_{ijk} – стоимость k-й версии j-й альтернативы i-го модуля; A_{ij} – событие, соответствующее отклонению результата j-й альтернативы i-го модуля; B_{ij} – событие, соответствующее принятию корректного результата j-й альтернативы i-го модуля; p_1 – вероятность того, что следующая альтернатива не будет вызвана, несмотря на отказ текущей альтернативы; p_2 – вероятность неверной оценки корректного результата; p_3 – вероятность принятия неверного результата за корректный.

Решение задачи формирования программной системы согласно схеме блока восстановления с согласованием, облачающейся высокой надежностью и низкой стоимостью, при этом учитывая совместимость входящих в его состав программных компонент, возможно с помощью методов дискретной оптимизации или методов многоатрибутивного принятия решений [3].

Заключение

Применение блока восстановления с согласованием позволяет достичь высокой надежности программной системы. Использование программной избыточности согласно данному подходу гарантирует высокий уровень надежности, то требует дополнительных ресурсов. Это проводит к необходимости решения задачи оптимизации, позволяющей обеспечить высокую надежность при заданных ограничениях.

При создании избыточной программной системы целесообразно введение в ее состав программных компонент не только собственной разработки, но приобретенных программных компонент – *COTS*-компонент. Однако при их интеграции в состав программной системы необходимо учитывать совместимость программных компонент.

В работе представлена модель оптимизации, предназначенная для решения обеих указанных проблем при формировании программной системы. Предложенный математический аппарат обладает большим потенциалом к использованию при создании высоконадежных программных систем.

Список литературы

1. Ковалев, И. В. Архитектурная надежность программного обеспечения информационно-управляющих систем: монография / И. В. Ковалев, Р. Ю. Царев, Д. В. Капулин; Краснояр. гос. аграр. ун-т. – Красноярск, 2011. – 182 с.

2. 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; URL: http://www.science-education.ru/121-18871 (дата обращения: 30.04.2015).

3. Царев, Р. Ю. Методология многоатрибутивного формирования мультиверсионного программного обеспечения сложных систем управления и обработки информации: монография / Р. Ю. Царев; Краснояр. гос. аграр. ун-т. – Красноярск, 2011. – 210 с.

4. Царев, Р. Ю. Мультиверсионное программное обеспечение. Алгоритмы голосования и оценка надежности: монография / Р. Ю. Царев, А. В. Штарик, Е. Н. Штарик. – Красноярск: Сибирский федеральный университет, 2013. – 120 с.

5. Bali, S., Gupta, A., Dinesh Kumar, U. Fuzzy multi-objective build-or-buy approach for component selection of fault tolerant modular software system under consensus recovery block scheme (2012) Advances in Intelligent and Soft Computing, 130 AISC (VOL. 1), pp. 1025-1036.
6. Berman, O., Kumar, U.D. Optimization models for recovery block schemes (1999) European Journal of Operational Research, 115 (2), pp. 368-379.
7. Cortellessa, V., Marinelli, F., Potena, P. An optimization framework for "build-or-buy" decisions in software architecture (2008) Computers and Operations Research, 35 (10), pp. 3090-3106.
8. Fiondella, L., Zeephongsekul, P. Recovery block fault tolerance considering correlated failures (2014) Proceedings - Annual Reliability and Maintainability Symposium, art. no. 6798525.
9. Nan, Z., Jiamin, W. Reliability analysis of COTS-based software system (2014) International Journal of Multimedia and Ubiquitous Engineering, 9 (8), pp. 73-84.
10. Randell, B., Jie, X., The Evolution of the Recovery Block Concept (1995) Software Fault Tolerance, Michael R. Lyu, editor, Wiley, pp. 1–21.

Рецензенты:

Бронов С. А., д.т.н., профессор, руководитель научно-учебной лаборатории систем автоматизированного проектирования кафедры систем искусственного интеллекта Сибирского федерального университета, г. Красноярск.

Носков М. В., д.ф.-м.н., профессор, заместитель директора по научной работе Института космических и информационных технологий Сибирского федерального университета, г. Красноярск.