Оригинальная статья
Надежность, том 18, №3, 2018 DOI: 10.21683/1729-2646-2018-18-3-3-9

Анализ дерева отказов в среде программирования R. Учет отказов по общей причине

Александр В. Антонов, *АО РАСУ, Москва, Россия*
Евгений Ю. Галивец, *АО РАСУ, Москва, Россия*
Валерий А. Чепурко, *АО РАСУ, Москва, Россия*
Алексей Н. Черняев, *АО РАСУ, Москва, Россия*

Александр В. Антонов

Евгений Ю. Галивец

Валерий А. Чепурко

Алексей Н. Черняев

Резюме. Цель. *Данная статья является продолжением работы [1], в которой для анализа дерева отказов (АДО) предлагается к применению язык программирования R. В [1] рассмотрены три примера: расчет дерева отказов (ДО) по известным вероятностям, расчет динамического ДО по известным распределениям наработок на отказ составляющих систему элементов. В последнем примере выполняется АДО для систем с элементами, которые описываются различными моделями функционирования и обслуживания. АДО – это один из основных методов анализа надежности сложных технических систем. Для его проведения часто применяются коммерческие программные средства, такие как Saphire, Risk Spectrum, PTC Windchill Quality, Арбитр и т.д. Практически каждое программное средство позволяет производить расчет надежности сложных систем с учетом возможного возникновения отказов по общей причине (ООП). ООП представляют собой зависимые отказы группы из нескольких элементов, происходящие одновременно или в течение короткого промежутка времени (т.е. почти одновременно), вследствие действия одной общей причины (например, резкое изменение климатических условий эксплуатации, затопление помещения эксплуатации и пр.). Зависимый отказ – это множественный отказ нескольких элементов системы, вероятность которого не может быть выражена просто как произведение вероятностей безусловных отказов отдельных элементов. Для расчета вероятностей ООП существуют несколько различных общепринятых моделей: модель греческих букв, альфа-, бета-фактора и различные их вариации. Наиболее простой с точки зрения моделирования зависимых отказов и проведения дальнейших расчетов надежности является модель бета-фактора. Остальные модели при моделировании подразумевают комбинаторный перебор зависимых событий в группе из n событий, который при большом числе n становится трудозатратным. Поэтому в указанных выше программных средствах существуют определенные ограничения на n, при выходе за пределы которых вероятность ООП рассчитывается приближенно. В пакете FaultTree языка R обозначенные выше модели ООП в современной версии отсутствуют, поэтому все зависимые отказы приходится моделировать самостоятельно, что несложно при малом числе зависимых событий и полезно для понимания сути различных моделей ООП. В статье для выбранной структуры подробно разобрана процедура моделирования зависимых отказов для модели альфа- и бета-фактора.* **Цель** *данной статьи состоит в подробном анализе методик альфа- и бета-фактора для некоторой структуры, в демонстрации процедуры создания ДО с учетом ООП с помощью пакета FaultTree языка программирования R.* **Методы.** *Для выполнения расчетов и демонстрации возможностей АДО применялись скрипты пакета FaultTree языка программирования R.* **Выводы.** *В статье подробно разобраны два примера. В первом примере для выбранной структурной схемы, содержащей две группы элементов, подверженных зависимым отказам, применяется модель альфа-фактора. Во втором примере применяется модель бета-фактора. Указаны недостатки современной версии пакета FaultTree. К основным недостаткам следует отнести отсутствие некоторых основных логических вентилей.*

Ключевые слова: *дерево отказов, анализ дерева отказов, отказ по общей причине, отказ по всевозможным причинам, независимые отказы, зависимые отказы, несовместные события, альфа-фактор, бета-фактор.*

Формат цитирования: *Антонов А.В., Галивец Е.Ю., Чепурко В.А., Черняев А.Н. Анализ дерева отказов в среде программирования R. Учет отказов по общей причине // Надежность. 2018. Т. 18, № 3. С. 3-9. DOI: 10.21683/1729-2646-2018-18-3-3-9*

Введение

Данная статья продолжает работу [1], посвященную обзору возможностей пакета FaultTree, разработанного для среды программирования R. R – язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом, разработанный в рамках проекта GNU. R поддерживает широкий спектр статистических и численных методов и обладает хорошей расширяемостью с помощью пакетов. Пакеты представляют собой библиотеки для работы специфических функций и подпрограмм или специальных областей применения. В статье продолжается анализ возможностей пакета создания, вычисления и вывода деревьев отказов – пакета FaultTree в плане учета отказов по общей причине (ООП).

Метод анализа дерева отказов (fault tree analysis, FTA) – это метод анализа надежности сложных систем, в котором отказы системы анализируются с помощью методов булевой алгебры, объединяя последовательность нижестоящих событий (отказов низшего уровня), которые приводят к отказу всей системы. При этом определяются цепочки случайных событий, при которых система может выйти из строя, выявляются способы уменьшения рисков и определяются частоты системных отказов. При этом в самых простых случаях деревья отказов формируют независимые события. Однако возможны ситуации, когда отказы возникают по общей причине, т.е. являются зависимыми от некоторого внутреннего или внешнего фактора. К внутренним факторам относят общие конструктивные, технологические и прочие внутренние причины, к внешним – воздействия природных явлений и/или деятельности человека [2-4].

Непосредственно для расчета вероятностей ООП широко применяются различные математические модели, связывающие линейно вероятности зависимого отказа подмножества элементов, подверженных ООП с вероятностью отказа по всевозможным (total) причинам. При этом отказ по всевозможным причинам по сути является полной группой, в которую входят, независимые отказы каждого элемента, ООП двух, трех и т.д. элементов. Достаточно простая, с точки зрения реализации, модель бета-фактора подразумевает, что в множестве элементов, подверженных ООП, отказы могут быть только двух типов: независимые одиночные отказы элементов и зависимые отказы всей группы ООП, происходящие одновременно или почти одновременно. В этом случае в дерево отказов несложно самостоятельно ввести все эти события. При этом желательно учесть, что они должны быть несовместными, т.е. связывающие выше эти события логические операции должны учитывать этот факт. При относительно малых вероятностях отказа можно применять оператор «или», при этом ошибка расчета будет мала.

Модель бета-фактора является частным случаем более общих моделей греческих букв и альфа-фактора. Отметим, что у последней модели существует несколько модификаций. Основное отличие обобщенных моделей от модели бета-фактора состоит в том, что зависимым отказам могут быть подвержены любые подмножества из множества элементов, подверженных ООП. При этом выбор таких подмножеств должен быть обоснован тем, что их сочетание должно сопровождаться отказом системы. Понятно, что здесь возникает комбинаторный перебор таких ситуаций, который при малом объеме множества (два, три элемента) возможен вручную, однако при больших объемах необходимо пользоваться вычислительными средствами, а точнее специализированными программными продуктами – Windchill PTC, Risk Spectrum, Арбитр и т.д. При этом и в самих программных средствах существуют некоторые предельные ограничения на объемы множеств, при невыполнении которых расчет ведется приближенно. Связано это с тем, что по мере увеличения объема множества элементов, подверженных ООП, несоизмеримо возрастают вычислительные затраты.

Что касается пакета FaultTree, то в его современной версии пока отсутствуют модели расчета ООП, поэтому в обобщенных моделях все переборы необходимо выполнять вручную. В связи с этим возникают другие проблемы, связанные с недостатком необходимых логических операций и (или) категорий событий, которые будут отражены в данной статье.

Рассмотрим некоторые основные возможности учета ООП с помощью пакета FaultTree.

Учет отказов по общим причинам

Для демонстрации возможностей учета ООП рассмотрим четыре различные модели: бета-фактора, альфа-фактора (с шахматным и не шахматным порядком испытаний) и модели греческих букв [5-7]. В качестве исходной схемы рассмотрим цепь (см. рисунок 1), представленную в работе [1].

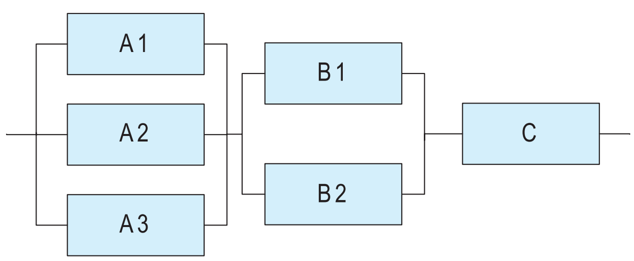

Рисунок 1 – Схема системы

Предположим, что отказать по общим причинам могут элементы группы А (А1, А2, А3) и элементы группы В (В1, В2). Введем следующие обозначения:

$I_1(A)$, $I_2(A)$, $I_3(A)$ – независимые (одиночные) отказы элементов группы А;

$C_{12}(A)$, $C_{23}(A)$, $C_{13}(A)$ – ООП ровно двух элементов группы А;

$C_{123}(A)$ – ООП всех трех элементов группы А;

$I_1(B)$, $I_2(B)$– независимые отказы элементов группы B;
$C_{12}(B)$ – ООП всех элементов группы B;
$F(C)$ – отказ элемента C.

Основная параметрическая модель ООП-анализа определяет следующие события:

$$\begin{aligned}1_t(A) &= I_1(A) + C_{12}(A) + C_{13}(A) + C_{123}(A):\\ 2_t(A) &= I_2(A) + C_{12}(A) + C_{23}(A) + C_{123}(A);\\ 3_t(A) &= I_3(A) + C_{13}(A) + C_{23}(A) + C_{123}(A);\\ 1_t(B) &= I_1(B) + C_{12}(B);\\ 2_t(B) &= I_2(B) + C_{12}(B).\end{aligned} \tag{1}$$

К примеру, первое событие будет означать отказ по всевозможным (total) причинам, связанный с отказом первого элемента группы A. Обозначим вероятности этих событий:

$$\begin{aligned}Q_t(A) &= \Pr(1_t(A)) = \Pr(2_t(A)) = \Pr(3_t(A));\\ Q_1(A) &= \Pr(I_1(A)) = \Pr(I_2(A)) = \Pr(I_3(A));\\ Q_2(A) &= \Pr(C_{12}(A)) = \Pr(C_{13}(A)) = \Pr(C_{23}(A));\\ Q_3(A) &= \Pr(C_{123}(A));\\ Q_t(B) &= \Pr(1_t(B)) = \Pr(2_t(B));\\ Q_1(B) &= \Pr(I_1(B)) = \Pr(I_2(B));\\ Q_2(B) &= \Pr(C_{12}(B)).\end{aligned} \tag{2}$$

Равенства (2) обоснованы тем фактом, что по предположению все элементы одной и той же группы идентичны и эксплуатируются в одинаковых условиях, а, следовательно, их показатели надежности одинаковы.

В силу несовместности событий в правой части каждого из уравнений (1) получаем:

$$\begin{aligned}Q_t(A) &= Q_1(A) + 2Q_2(A) + Q_3(A);\\ Q_t(B) &= Q_1(B) + Q_2(B).\end{aligned} \tag{3}$$

Вероятности правых частей уравнений (3) определяются по-разному в зависимости от моделей.

Модель греческих букв

Так, для модели греческих букв справедливо следующее предположение:

$$Q_k^{(m)} = \left(\frac{1-p_{k+1}}{C_{m-1}^{k-1}}\prod_{i=1}^{k} p_i\right)Q_t,\ p_1 = 1,\dots,p_{m+1} = 0. \tag{4}$$

В нашем случае, если обозначить: $p_2 = \beta$, $p_3 = \gamma$, из (4) легко получается:

$$\begin{aligned}Q_1^{(3)}(A) &= (1-\beta(A))Q_t(A);\\ Q_2^{(3)}(A) &= \frac{1}{2}\beta(A)(1-\gamma(A))Q_t(A);\\ Q_3^{(3)}(A) &= \beta(A)\gamma(A)Q_t(A);\\ Q_1^{(2)}(B) &= (1-\beta(B))Q_t(B);\\ Q_2^{(2)}(B) &= \beta(B)Q_t(B).\end{aligned} \tag{5}$$

Модель альфа-фактора (не шахматный порядок испытаний) (Not-staggered Testing)

В этом случае общая формула для вероятностей такова:

$$Q_k^{(m)} = \left(\frac{k\alpha_k^{(m)}}{C_{m-1}^{k-1}\alpha_t}\right)Q_t,\ \text{где}\ \alpha_t = \sum_{k=1}^{m} k\alpha_k^{(m)}. \tag{6}$$

Для групп из 3-х и 2-х событий, соответственно, получим:

$$\begin{aligned}Q_1^{(3)}(A) &= \frac{\alpha_1^{(3)}}{\alpha_t^{(3)}}Q_t(A);\\ Q_2^{(3)}(A) &= \frac{\alpha_2^{(3)}}{\alpha_t^{(3)}}Q_t(A);\\ Q_3^{(3)}(A) &= \frac{3\alpha_3^{(3)}}{\alpha_t^{(3)}}Q_t(A);\ \textit{где}\ \alpha_t^{(3)} = \alpha_1^{(3)} + 2\alpha_2^{(3)} + 3\alpha_3^{(3)};\\ Q_1^{(2)}(B) &= \frac{\alpha_1^{(2)}}{\alpha_t^{(3)}}Q_t(B);\\ Q_2^{(2)}(B) &= \frac{2\alpha_2^{(2)}}{\alpha_t^{(2)}}Q_t(B);\ \textit{где}\ \alpha_t^{(2)} = \alpha_1^{(2)} + 2\alpha_2^{(2)}.\end{aligned} \tag{7}$$

Модель альфа-фактора (шахматный порядок испытаний) (Staggered Testing)

В этом случае общая формула для вероятностей такова:

$$Q_k^{(m)} = \left(\frac{\tilde\alpha_k^{(m)}}{C_{m-1}^{k-1}}\right)Q_t,\ \text{где}\ \sum_{k=1}^{m}\tilde\alpha_k^{(m)} = 1. \tag{8}$$

Для групп из 3-х и 2-х событий, соответственно, получим:

$$\begin{aligned}Q_1^{(3)}(A) &= \alpha_1^{(3)}Q_t(A);\\ Q_2^{(3)}(A) &= \frac{1}{2}\tilde\alpha_2^{(3)}Q_t(A);\\ Q_3^{(3)}(A) &= \tilde\alpha_3^{(3)}Q_t(A)\ \textit{при этом}\ \tilde\alpha_1^{(3)} + \tilde\alpha_2^{(3)} + \tilde\alpha_3^{(3)} = 1;\\ Q_1^{(2)}(B) &= \tilde\alpha_1^{(2)}Q_t(B);\\ Q_2^{(2)}(B) &= \tilde\alpha_2^{(2)}Q_t(B)\ \textit{при этом}\ \tilde\alpha_1^{(2)} + \tilde\alpha_2^{(2)} = 1.\end{aligned} \tag{9}$$

Модель бета-фактора

Одна из самых простых моделей учета ООП выглядит следующим образом:

$$Q_k^{(m)} = \begin{cases}(1-\beta)Q_t, & k = 1;\\ 0, & 1 < k < m;\\ \beta Q_t, & k = m.\end{cases} \tag{10}$$

В нашем случае получается:

$$\begin{aligned}Q_1^{(3)}(A) &= (1-\beta(A))Q_t(A);\\ Q_2^{(3)}(A) &= 0;\\ Q_3^{(3)}(A) &= \beta(A)Q_t(A);\\ Q_1^{(2)}(B) &= (1-\beta(B))Q_t(B);\\ Q_2^{(2)}(B) &= \beta(B)Q_t(B).\end{aligned} \tag{11}$$

Нетрудно убедиться, что при подстановке (5), (7), (9), (11) в (3) получается тождество $Q_t(A) = Q_t(A)$, $Q_t(B) = Q_t(B)$, впрочем, это безусловно будет выполняться и при больших m. Таким образом, различие подходов моделей состоит лишь в различном понимании того, как соотносятся между собой вероятности $Q_1^{(m)}, Q_2^{(m)}, \ldots, Q_m^{(m)}$. При этом часто разные модели могут давать достаточно близкие результаты. Для этого существуют формулы перехода [5] см. табл. А-2–А-4 приложения А. Кроме этого в том же источнике [4] в табл. 5.11 стр. 75 можно найти справочную статистическую информацию о параметрах $\tilde{\alpha}_k^{(m)}$ для модели альфа-фактора (8). Так, для параллельной цепи из двух элементов B1, B2 выборочные медианы параметров (50% точки) равны соответственно

$$med\left(\tilde{\alpha}_1^{(2)}\right) = 0,953,\ med\left(\tilde{\alpha}_2^{(2)}\right) = 0,047. \qquad (12)$$

Для подцепи А из трех элементов А1, А2, А3–

$$med\left(\tilde{\alpha}_1^{(3)}\right) = 0,9500,\ med\left(\tilde{\alpha}_2^{(3)}\right) = 0,0242,$$
$$med\left(\tilde{\alpha}_3^{(3)}\right) = 0,0258. \qquad (13)$$

Возьмем в качестве значений параметров модели (8) эти числа. Вероятности (9) будут равны

$$\begin{aligned} &Q_1^{(3)}(A) = 0,9500 \cdot Q_t(A);\\ &Q_2^{(3)}(A) = 0,0121 \cdot Q_t(A);\\ &Q_3^{(3)}(A) = 0,0258 \cdot Q_t(A);\\ &Q_1^{(2)}(B) = 0,953 \cdot Q_t(B);\\ &Q_2^{(2)}(B) = 0,047 \cdot Q_t(B). \end{aligned} \qquad (14)$$

Можно воспользоваться таблицами перехода, но нетрудно догадаться, что в модели (5)

$$\beta(A) = 0,05,\ \gamma(A) = \frac{0,0258}{\beta(A)} = 0,516,\ \beta(B) = 0,047.$$

В модели альфа-фактора с (не шахматным порядком испытаний) после несложных преобразований получим:

$$\alpha_1^{(3)} = 0,95,\ \alpha_2^{(3)} = 0,0121,\ \alpha_3^{(3)} = 0,0086,\ \alpha_t^{(3)} = 1,$$

$$\alpha_1^{(2)} = 0,953,\ \alpha_2^{(2)} = 0,0235,\ \alpha_t^{(2)} = 1.$$

При выведенных значениях параметров результаты обеих моделей альфа-фактора и модели греческих букв дадут идентичный результат. Для достаточно грубой, а с другой стороны и более простой, модели бета-фактора результаты будут иными, поскольку в модели бета-фактора используется всего один входной параметр. Тем не менее, возьмем его таким же, как соответствующая греческая буква – 0,05.

Теперь приступим к расчетам. Для упрощения дерева отказов не будем рассматривать различные модели надежности для элементов, а будем считать, что вероятности отказов элементов *A*, *B* и *C* равны, соответственно

$$Q_t(A) = 0.3,\ Q_t(B) = 0,2,\ Q_t(C) = 0,1. \qquad (15)$$

Вероятность отказа без учета ООП будет равна:

$$Q_S = 1 - \left(1 - Q_t^3(A)\right)\left(1 - Q_t^2(B)\right)\left(1 - Q_t(C)\right) = 0,1593. \qquad (16)$$

Проведем расчеты с учетом ООП. Наша цепь откажет при следующих сочетаниях событий, представленных в виде восьми минимальных сечений:

$$\begin{gathered} \{I_1(A) \cap I_2(A) \cap I_3(A)\}, \{I_1(A) \cap C_{23}(A)\},\\ \{I_2(A) \cap C_{13}(A)\}, \{I_3(A) \cap C_{12}(A)\}, \{C_{123}(A)\}, \\ \{I_1(B) \cap I_2(B)\}, \{C_{12}(B)\}, \{F(C)\}. \end{gathered} \qquad (17)$$

Составим скрипт расчета. В отличие от специализированных пакетов, упомянутых выше, в современной версии пакета учет ООП не выполняется, поэтому все события (17) приходится разрабатывать и вводить самостоятельно. Заметим, что в (17) есть группа несовместных (а, значит, зависимых) сечений, к примеру, первое и второе, первое и третье и т.д. Есть и группа независимых сечений, к примеру, $\{I_1(A) \cap I_2(A) \cap I_3(A)\}$ и $\{C_{12}(B)\}$, т.е. сечений, принадлежащих различным группам ООП. Для корректного расчета вероятностей отказа этой группы необходимо применять специализированный логический узел «или», который рассчитывает вероятность суммы несовместных событий. С другой стороны, можно ввести дополнительный тип для группы несовместных событий, содержащихся в одной группе ООП. И, наверное, самый оптимальный вариант состоит в разработке модуля учета ООП, который, возможно, без графического отображения в дереве отказов, при выделении групп событий ООП и выборе соответствующей модели будет самостоятельно и корректно рассчитывать показатели надежности. К сожалению, эти возможности пока не реализованы в R. Поэтому для расчетов будем использовать обычное «или».

Пример 1. ООП. Модель альфа-фактора.

```
  library(FaultTree)
  tree4 <- ftree.make(type="or",
name="Пример 4.", name2="ООП")
  tree4 <- addLogic(tree4, at=1,
type="and", name="I1(A)*I2(A)*I3(A)")
  tree4 <- addLogic(tree4, at=2,
type="inhibit", name="Независимый",
name2="отказ Ai")
  tree4 <- addProbability(tree4, at=3,
prob=.95, name="Параметр", name2="модели")
  tree4 <- addProbability(tree4, at=3,
prob=.3, name="Отказ Ai", name2="(total)")
  tree4 <- addDuplicate(tree4, at=2, dup_
id=3)
  tree4 <- addDuplicate(tree4, at=2, dup_
id=3)
  tree4 <- addLogic(tree4, at=1,
type="and", name="Ii(A)*Cjk(A)")
```

```
  tree4 <- addDuplicate(tree4, at=12, dup_
id=3)
  tree4 <- addLogic(tree4, at=12,
type="inhibit", name="ООП", name2="отказ
Aj, Ak")
  tree4 <- addProbability(tree4,
at=16, prob=.0121, name="Параметр",
name2="модели")
  tree4 <- addProbability(tree4, at=16,
prob=.3, name="Отказ Ai", name2="(total)")
  tree4 <- addDuplicate(tree4, at=1, dup_
id=12)
  tree4 <- addDuplicate(tree4, at=1, dup_
id=12)
  tree4 <- addLogic(tree4, at=1,
type="inhibit", name="ООП C123(A)",
name2="отказ A1,A2,A3")
  tree4 <- addProbability(tree4,
at=33, prob=.0258, name="Параметр",
name2="модели")
  tree4 <- addProbability(tree4, at=33,
prob=.3, name="Отказ Ai", name2="(total)")
  tree4 <- addLogic(tree4, at=1,
type="and", name="I1(B)*I2(B)")
  tree4 <- addLogic(tree4, at=36,
type="inhibit", name="Независимый",
name2="отказ Bi")
  tree4 <- addProbability(tree4, at=37,
prob=.953, name="Параметр", name2="модели")
  tree4 <- addProbability(tree4, at=37,
prob=.2, name="Отказ Bi", name2="(total")
  tree4 <- addDuplicate(tree4, at=36, dup_
id=37)
  tree4 <- addLogic(tree4, at=1,
type="inhibit", name="ООП C12(B)",
name2="отказ B1,B2")
  tree4 <- addProbability(tree4, at=43,
prob=.047, name="Параметр", name2="модели")
  tree4 <- addProbability(tree4, at=43,
prob=.2, name="Отказ Bi", name2="(total)")
  tree4 <- addProbability(tree4, at=1,
prob=.1, name="Отказ C", name2="(total)")
  tree4 <- ftree.calc(tree4)
  ftree2html(tree4, write_file=TRUE)
  browseURL("tree4.html")
```

Особых комментариев по этому скрипту мы делать не будем. Обратим внимание на 4-ю, 11-ю, … строки. При добавлении логического элемента здесь применяется вентиль запрета– inhibit. Как известно [5-7], в этом случае выходное событие происходит, если происходит оба входных события, одно из которых условное. Роль условия играет коэффициент модели альфа-, бета-фактора или модели греческих букв, так как на самом деле эти коэффициенты играют роль условных вероятностей.

Для расчета модели бета-фактора, казалось бы, необходимо было сделать две незначительные правки. В 12-й строке указать вероятность равную 0, а в 17-й– 0,05. Однако в этом случае при расчете дерева отказов возникает ошибка, связанная с тем, что одна из вероятностей равна 0. Скорее всего, в будущем эта ошибка будет исправлена. Сейчас же, как минимум, возможны два подхода. Один из них – задать нулевую вероятность крайне малой. Другой – убрать ветви с нулевой вероятностью. Следующий пример как раз демонстрирует этот подход.

Пример 2. ООП. Модель бета-фактора.

```
  library(FaultTree)
  tree4 <- ftree.make(type="or",
name="Пример 4.», name2=»ООП»)
  tree4 <- addLogic(tree4, at=1,
type="and", name="I1(A)*I2(A)*I3(A)")
  tree4 <- addLogic(tree4, at=2,
type="inhibit", name="Независимый»,
name2=»отказ Ai»)
  tree4 <- addProbability(tree4, at=3,
prob=.95, name="Параметр», name2=»модели»)
  tree4 <- addProbability(tree4, at=3,
prob=.3, name="Отказ Ai», name2=»(total)»)
  tree4 <- addDuplicate(tree4, at=2, dup_
id=3)
  tree4 <- addDuplicate(tree4, at=2, dup_
id=3)
  tree4 <- addLogic(tree4, at=1,
type="inhibit", name="ООП C123(A)»,
name2=»отказ A1,A2,A3»)
  tree4 <- addProbability(tree4, at=12,
prob=.05, name="Параметр», name2=»модели»)
  tree4 <- addProbability(tree4, at=12,
prob=.3, name="Отказ Ai», name2=»(total)»)
  tree4 <- addLogic(tree4, at=1,
type="and", name="I1(B)*I2(B)")
  tree4 <- addLogic(tree4, at=15,
type="inhibit", name="Независимый»,
name2=»отказ Bi»)
  tree4 <- addProbability(tree4, at=16,
prob=.953, name="Параметр», name2=»модели»)
  tree4 <- addProbability(tree4, at=16,
prob=.2, name="Отказ Bi», name2=»(total»)
  tree4 <- addDuplicate(tree4, at=15, dup_
id=16)
  tree4 <- addLogic(tree4, at=1,
type="inhibit", name="ООП C12(B)»,
name2=»отказ B1,B2»)
  tree4 <- addProbability(tree4, at=22,
prob=.047, name="Параметр», name2=»модели»)
  tree4 <- addProbability(tree4, at=22,
prob=.2, name="Отказ Bi», name2=»(total)»)
  tree4 <- addProbability(tree4, at=1,
prob=.1, name="Отказ C», name2=»(total)»)
  tree4 <- ftree.calc(tree4)
  ftree2html(tree4, write_file=TRUE)
  browseURL("tree4.html")
```

Проведем расчеты аналитически. Вначале рассчитаем модели альфа-фактора и греческих букв. Подбором коэффициентов моделей мы добились одинаковых результатов. В силу условий (2) и независимости событий точная вероятность отказа по всем причинам (как по общим причинам, так и независимо друг от друга) будет равна

$$Q_{S(CCF)} = 1 - Pr\left(\overline{A} \cap \overline{B} \cap \overline{C}\right) = 1 - Pr\left(\overline{A}\right) Pr\left(\overline{B}\right) Pr\left(\overline{C}\right), \quad (18)$$

где события $\overline{A}$ ={не отказали элементы группы А}, $\overline{B}$ = {не отказали элементы группы В}, $\overline{C}$ = {не отказали элементы группы С}.

Т.к. независимые отказы и отказы по общим причинам несовместны, а значит зависимы между собой, то

$$\begin{cases} Pr\left(\overline{A}\right) = 1 - Q_1^3(A) - 3Q_1(A)Q_2(A) - Q_3(A), \\ Pr\left(\overline{B}\right) = 1 - Q_1^2(B) - Q_2(B). \end{cases} \quad (19)$$

Численное значение $Q_{S(CCF-\alpha)}$ = 0,170350. Расчетное приближенное значение $Q_{S(CCF-\alpha)}$ = 0,16981 – рисунок 2. На рисунке 2 приведено неполное дерево отказов с некоторыми «свернутыми» ветвями в силу его громоздкости.

Логический узел «или» не учитывает факт зависимости минимальных сечений в (17) и вычисляет $Pr\left(\overline{A}\right)$ и $Pr\left(\overline{B}\right)$ по следующим формулам:

$$\begin{cases} Pr\left(\overline{A}\right) = \left(1 - Q_1^3(A)\right)\left(1 - Q_1(A)Q_2(A)\right)^3\left(1 - Q_3(A)\right), \\ Pr\left(\overline{B}\right) = \left(1 - Q_1^2(B)\right)\left(1 - Q_2(B)\right). \end{cases} \quad (20)$$

Для модели бета-фактора (10) точная формула расчета вероятностей $Pr\left(\overline{A}\right)$ и $Pr\left(\overline{B}\right)$ будет выглядеть несколько проще:

$$\begin{cases} Pr\left(\overline{A}\right) = 1 - Q_1^3(A) - Q_3^*(A), \\ Pr\left(\overline{B}\right) = 1 - Q_1^2(B) - Q_2(B). \end{cases} \quad (21)$$

Приближенная формула:

$$\begin{cases} Pr\left(\overline{A}\right) = \left(1 - Q_1^3(A)\right)\left(1 - Q_3^*(A)\right), \\ Pr\left(\overline{B}\right) = \left(1 - Q_1^2(B)\right)\left(1 - Q_2(B)\right). \end{cases} \quad (22)$$

В (21) и (22) $Q_3^*(A) = 0.05Q_t(A)$. Точное и приближенное значения $Q_{S(CCF-\beta)}$ = 0,17392 и 0,17333 соответственно. Приближенная вероятность совпадает с расчетной (рисунок 3).

Как и следовало ожидать, модель бета-фактора оказалась пессимистичнее.

На рисунке приведено дерево отказов со свернутыми ветвями в силу его громоздкости.

В заключение хотелось бы отметить несколько существенных недостатков современной версии пакета FaultTree в плане возможности его применения для расчетов надежности сложных систем.

Выбор логических операций (вентилей) для работы с событиями весьма ограничен. Так, отсутствуют модули «взаимоисключающее или», «приоритетное и», «отрицание» и т.д. Это существенно ограничивает спектр возможностей пакета.

В пакете нет возможности дублирования одного и того же базисного события в различных ветвях дерева. Скрипт addDuplicate() просто упрощает построение сложных деревьев событий, дублируя при этом ветви, структуры. При этом учет факта зависимости, несовместности событий невозможен. Невозможно «вручную», с помощью имеющихся логических операций, создать дерево, содержащее подобного рода события. Об этом сказано выше.

Для возможности учета ООП хотелось бы иметь дополнительные скрипты различных моделей – альфа-, бета-фактора, греческих букв и т.д.

Богатый инструментарий языка R позволяет делать более гибкие настройки вычислений и в отличие от жестких схем специализированных пакетов, самостоятельно выполнять определенные процедуры с исходными данными. И, конечно же, самое важное преимущество языка R состоит в том, что в отличие от специализированных пакетов, предназначенных для анализа дерева событий и только, в нем присутствуют несравнимо большие возможности для выполнения процедур анализа данных.

В связи с этим крайне необходимая доработка пакета Fault Tree в смысле устранения обозначенных выше не-

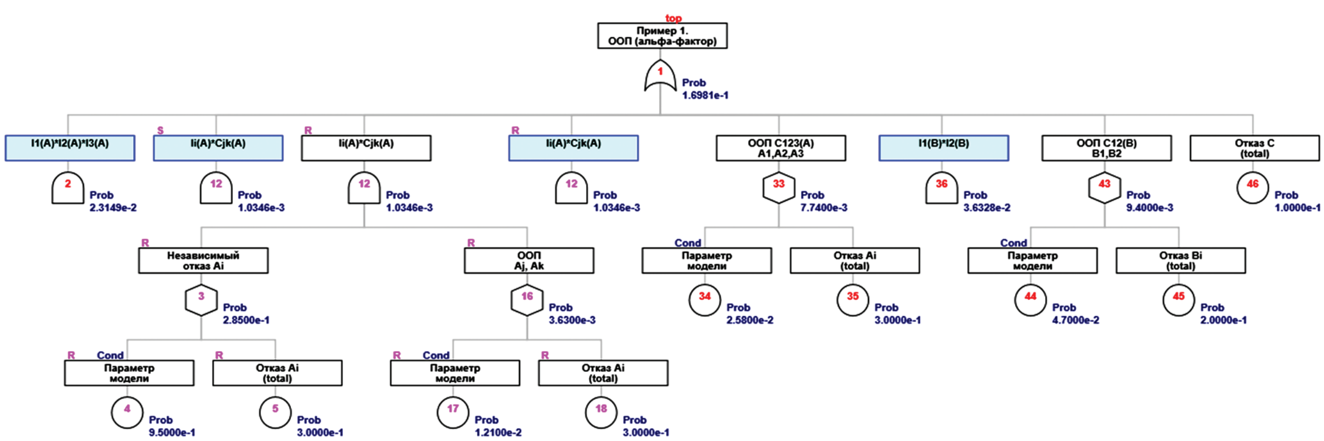

Рисунок 2 – Дерево отказов для примера 4 (модель альфа-фактора)

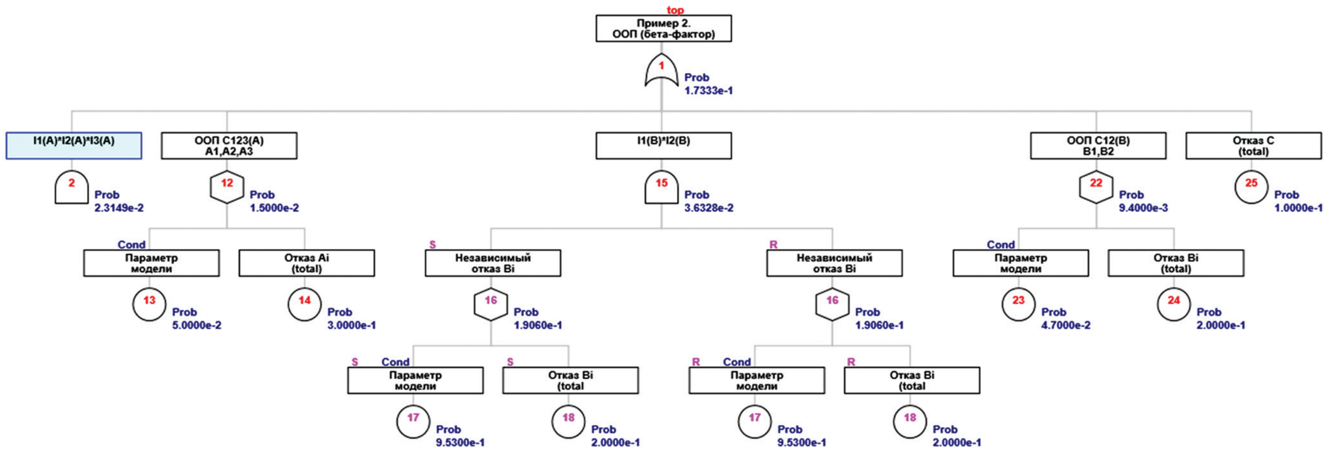

Рисунок 3 – Дерево отказов для примера 4 (модель бета-фактора)

достатков действительно даст специалистам мощный инструмент не только для анализа деревьев отказов, но и для проведения более продвинутого статистического анализа. Что касается дальнейшего развития пакета, то также в нем желательна доработка в плане написания функционала, обеспечивающего расчет различных показателей значимости по Бирнбауму, Веселы-Фасселу и т.д., проведения анализа неопределенности.

Заключение

Статья посвящена демонстрации возможностей активно развивающегося языка статистических вычислений R и его пакета FaultTree, связанных с построением и анализом деревьев отказов. Деревья отказов используются при анализе надежности сложных систем. В статье приведены и подробно разобраны некоторые модели учета ООП, изложены два примера. В первом примере проводится учет ООП по модели альфа-фактора. Второй пример посвящен модели бета-фактора. Указаны недостатки и необходимый вектор развития пакета FaultTree.

Библиографический список

1. Антонов, А.В. Анализ дерева отказов в среде программирования R [Текст] / А.В. Антонов, Е.Ю. Галивец, В.А. Чепурко, А.Н. Черняев // Надежность. – 2018. – № 1. – С. 4–13. doi:10.21683/1729-2646-2018-18-1-4-13

2. Перегуда, А.И. Математическая модель надежности компьютерных сетей [Текст] / А.И. Перегуда, А.А. Перегуда, Д.А. Тимашев // Надежность. – 2013. – № 4. – С. 18–43. doi:10.21683/1729-2646-2013-0-4-18-43

3. Алпеев, А.С. Надежность программного обеспечения управляющих систем и безопасность атомных станций [Текст] / А.С. Алпеев // Надежность. – 2015. – № 4. – С. 75–80. doi:10.21683/1729-2646-2015-0-4-75-80

4. Острейковский, В.А. Безопасность атомных станций. Вероятностный анализ [Текст] / В.А. Острейковский, Ю.В. Швыряев. – М.: «Физматлит», 2008. – 524 с.

5. Mosleh, A. Procedures Guidelines in Modeling Common Cause Failures in Probabilistic Risk Assessment [Текст] / A. Mosleh, et. al. – NUREG/CR-5485 (1998).

6. Программный комплекс автоматизированного структурно-логического моделирования и расчета надежности и безопасности АСУТП на стадии проектирования (ПК АСМ СЗМА) [Текст]: Техническая документация. – СПб.: ОАО «СПИК СЗМА», 2003.

7. Smith, C.L. Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) [Текст] / Version 8-Vol. 2 / C.L. Smith, S.T. Wood, W.J. Galyean, J.A. Schroeder, M.B. Sattison. – NUREG/CR-7039 INL/EXT-09-17009 (June 2011).

Сведения об авторах

Александр В. Антонов – доктор технических наук, профессор, главный эксперт отдела расчетных обоснований проектных решений АО РАСУ, e-mail: AlVlaAntonov@rasu.ru

Евгений Ю. Галивец – заместитель директора департамента – руководитель управления проектирования АО РАСУ, e-mail: EYGalivets@rasu.ru

Валерий А. Чепурко – кандидат физико-математических наук, доцент, главный специалист отдела расчетных обоснований проектных решений АО РАСУ, e-mail: VAChepurko@rasu.ru

Алексей Н. Черняев – кандидат технических наук, заместитель технического директора – директор департамента проектирования АО РАСУ, e-mail: AlNChernyaev@rasu.ru

Поступила: 19.03.2018