

## ГЕРТ-СЕТЕВОЙ АНАЛИЗ МУЛЬТИВЕРСИОННЫХ АРХИТЕКТУР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*В статье предлагается методика анализа мультиверсионных архитектур программного обеспечения, которая позволяет использовать алгоритмы и методы анализа сетей для определения надежности программного обеспечения, разработанного с использованием мультиверсионного подхода.*

*Ключевые слова: надежность, мультиверсионность, сети, методы оценки, методы анализа сетей.*

Программное обеспечение, являясь неотъемлемой составляющей коммерческих и специальных систем управления и обработки информации, проникает во многие области современной жизни, но, несмотря на столь широкое распространение, программное обеспечение едва когда-либо было совершенно. На сегодняшний день разработаны различные методы проектирования отказоустойчивого программного обеспечения. Среди них одним из наиболее перспективных является метод мультиверсионного проектирования.

Однако, учитывая сложность мультиверсионных систем обработки информации, множество параметров системы, которые могут изменяться во времени, прогнозировать время завершения задачи, а также надежность системы, основываясь на статических или детерминированных моделях систем или программ, достаточно сложно, а в некоторых случаях просто невозможно. Это обстоятельство представляется научной проблемой, выражающейся в необходимости поиска новых подходов к анализу надежности, а также временных характеристик работы программного обеспечения построенного на основе мультиверсионной архитектуры.

Одним из таких подходов является графоаналитический метод основанный на использовании ГЕРТ сетей. Основное достоинство этого подхода заключается в том, что он может быть успешно применен к решению практически любой задачи, и дает возможность составить формальные процедуры для определения качественных характеристик системы [1].

Опишем модель N-версионного программирования [2] (рис. 1) в виде ГЕРТ сети (рис. 2) и рассчитаем её основные характеристики, при условии, что количество мультиверсий  $N = 3$ , а остальные параметры описаны в табл. 1.

Таблица 1

### Характеристика операций

Ветвь	Вероятность что операция будет выполнена ( $p_i$ )	Тип распределения	Параметры (мс)
(1,4)	0,85	нормальное	$m=0,5$ $\sigma=0,1$
(2,4)	0,85	нормальное	$m=0,5$ $\sigma=0,1$
(3,4)	0,85	нормальное	$m=0,5$ $\sigma=0,1$
(4,выход)	0,99	нормальное	$m=2$ $\sigma=0,5$

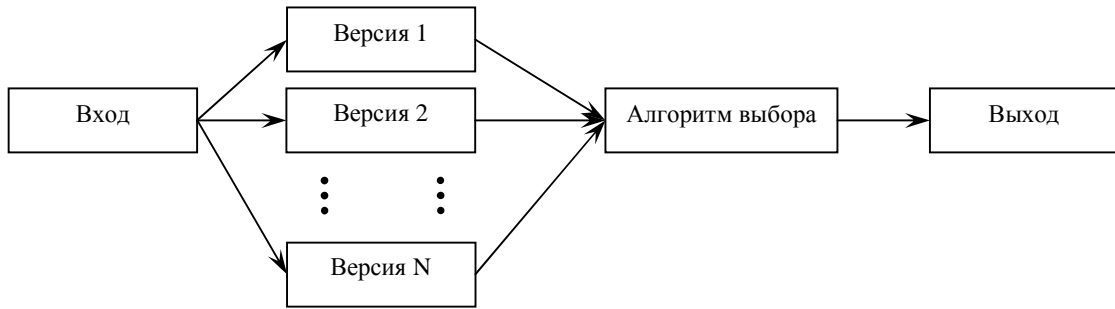


Рис. 1. Модель N-версионного программирования

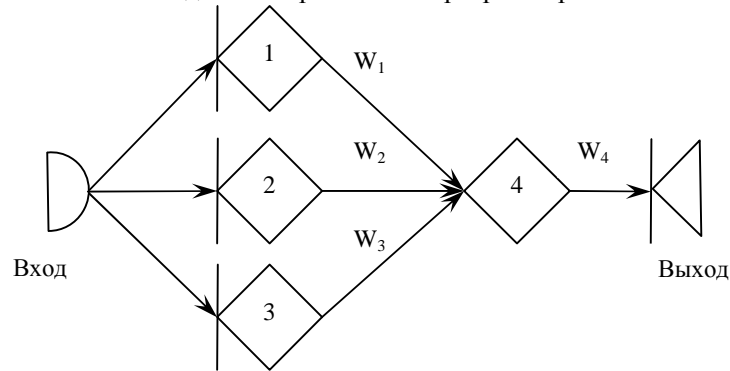


Рис. 2. Модель N-версионного программирования в виде ГЕРТ сети

Узлы стохастической сети (рис. 2) могут быть интерпретированы как состояния системы, а дуги — как переходы из одного состояния в другое. Такие переходы можно рассматривать как выполнение обобщенных операций, характеризующихся плотностью распределения, или функцией массы, и вероятностью выполнения.

Каждый внутренний узел стохастической сети выполняет две функции, одна из которых касается входа в узел, а другая — выхода. Обычно эти функции называют входной и выходной.

- Входная функция. Она определяет условие, при котором узел может быть выполнен.
- Выходная функция. Она определяет совокупность условий, связанных с результатом выполнения узла. Другими словами, с помощью выходной функции указывается, должны ли выполняться все операции, которым данный узел непосредственно предшествует, или только одна из них.

Отметим, что начальный узел сети выполняет только выходную функцию, в то время как конечный узел — только входную [3].

Рассмотрим виды входных функций:

1. AND-функция – узел активируется, если выполнены все дуги, входящие в него.
2. IOR-функция – узел активируется, если выполнена любая дуга, входящая в него.
3. EOR-функция – узел активируется, если выполнена любая дуга, входящая в него, при условии, что в данный момент времени может выполняться только одна дуга, входящая в данный узел.

Рассмотрим виды выходных функций:

1. Детерминированная функция – все дуги, выходящие из узла, выполняются, если узел активирован.
2. Стохастическая функция – ровно одна дуга, выходящая из узла, выполняется с заданной вероятностью, если узел активирован.

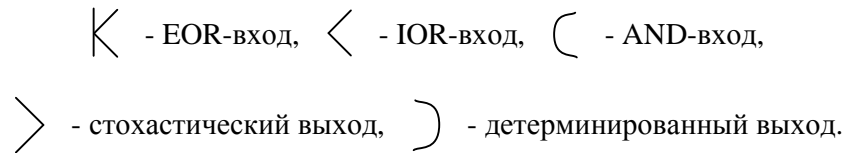


Рис. 3. Графическое обозначение входных и выходных функций ГЕРТ-сети

Комбинируя все входные и выходные функции (рис. 3), получаем шесть различных типов узлов. Активация узла означает, что система перешла в некоторое состояние и определяет множество возможных дальнейших действий. Одно или несколько действий начинают свое выполнение сразу после активации узла, являющегося их началом. Активация узла происходит, если его входная функция выполнена. После выполнения выходной функции активированного узла (начала выполнения соответствующей дуги) он становится неактивным [1].

Необходимым и достаточным условием функционирования мультиверсионного модуля является выполнение хотя бы одной мультиверсии. Вот почему узел 4 на рис.2 является узлом с IOR входом. Выбор данного типа узла связан с тем, что по определению этот узел активируется при выполнении любой дуги входящей в него.

Рассмотрим расчет сети содержащей IOR вход и детерминированный выход рис.4, состоящей из нескольких подсетей рис. 5

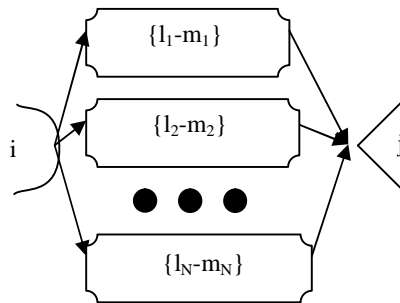


Рис. 4. IOR-вход

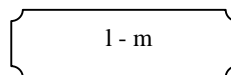


Рис. 5. Произвольная подсеть с начальным узлом 1 и конечным узлом m

Необходимо учитывать, что  $j$  – вычисляемый IOR-вход, имеющий стохастическое начало в узле  $i$ .  $\{11-m1\}$ ,  $\{12-m2\}$ , ...,  $\{1N-mN\}$  –  $N$  непересекающихся подсетей.

Пусть  $p_{ij}$  — вероятность того, что операция  $(i, j)$  будет выполнена при условии, что узел  $i$  выполнен, тогда имеем:

$$p_j = p_i \cdot [1 - (1 - p_{l1,m1k}) \cdot (1 - p_{l2,m2k}) \cdot \dots \cdot (1 - p_{lN,mNk})] \quad (1)$$

Учитывая что от входа до узла 4 находится 3 непересекающиеся подсети, в данном частном случае, используя выражение (1) получим:

$$p_4 = p_{\text{вход}} \cdot [1 - (1 - p_1) \cdot (1 - p_2) \cdot (1 - p_3)] = 1 - (1 - 0,85)^3 = 0,996625 \quad (2)$$

Итак, воспользовавшись правилом Мейсона для замкнутых потоковых графов получим эквивалентную W-функцию [3] для этой сети:

$$W_E(s) = W_{\Sigma}(s) \cdot W_4(s) = 0,9966e^{0,5 \cdot s + \frac{1}{2} \cdot 0,1^2 \cdot s^2} \cdot 0,99e^{2 \cdot s + \frac{1}{2} \cdot 0,5^2 \cdot s^2} = 0,9866e^{2,5 \cdot s + 0,13 \cdot s^2} \quad (3)$$

Легко проверить что в рассматриваемой задаче  $W_E(0) = 0.9866$ . Данную величину можно интерпретировать как вероятность безотказной работы мультиверсионного ПО. В свою очередь математическое ожидание времени работы равно:

$$m_{1E} = \frac{\partial M_E(s)}{\partial s} \Big|_{s=0} = \frac{\partial \left( \frac{W_E(s)}{W_E(0)} \right)}{\partial s} \Big|_{s=0} = 2.5 \text{ мс} \quad (4)$$

И дисперсия

$$m_{1E} = \frac{\partial^2 M_E(s)}{\partial s^2} \Big|_{s=0} = 6.51 \text{ мс}^2 \quad (5)$$

Следует заметить, что при расчет сети содержащей IOR вход и детерминированный выход, если все подсети рис.4 представляют собой мультиверсионные модули с одинаковой надежностью, то формула (1) может быть сведена к виду

$$p_j = p_i \cdot [1 - (1 - p_{1,m})^n] \quad (6)$$

Зависимость между числом мультиверсионных модулей и надежностью системы демонстрирует рис.6. На оси X обозначено количество мультиверсий, на оси Y – надежность системы для разного количества модулей с разной надежностью каждого модуля (от 30 до 80%)

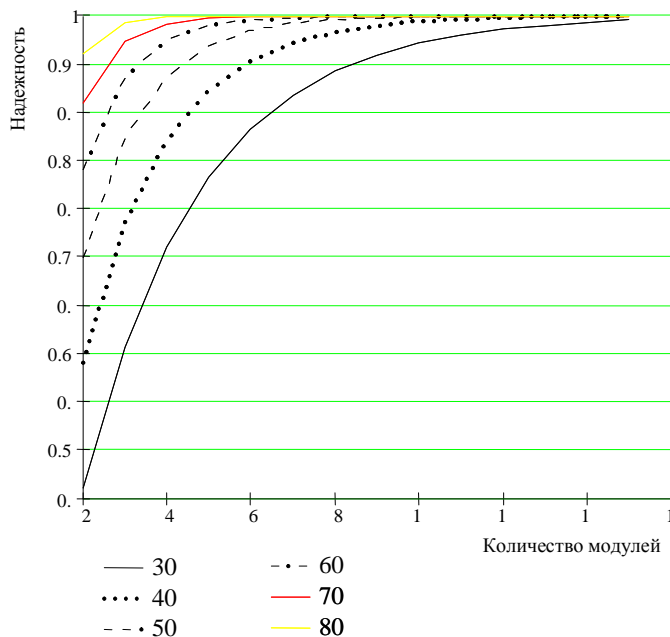


Рис. 6. Зависимость между числом мультиверсионных модулей и надежностью системы

Таким образом, графоаналитический метод основанный на использовании ГЕРТ сетей является перспективным, так как позволяет аналитически оценить качественные характеристики мультиверсионного ПО любой сложности без построения громоздких имитационных сред и комплексов программ. Кроме того, расчеты показали, что с увеличением числа модулей надежность системы растет, что подтверждает перспективность использования мультиверсионного подхода для повышения надежности ПО.

### Библиографический список

1. Ковалёв, П. В. Определение надежности мультиверсионного программного обеспечения с использованием методов анализа сетей / П. В. Ковалёв, А. Н. Лайков, С. Н. Гриценко // Вестник СибГАУ. 2009. № 1(22) в 2 частях. Часть 2. стр. 55-60.
2. Algirdas Avizienis, The Methodology of N-Version Programming, in R. Lyu, ed itor, Software Fault Tolerance, John Wiley & Sons, 1995.
3. Филлипс, Д., Гарсиа-Диаз, А. Методы анализа сетей / Д. Филлипс, А. Гарсиа-Диаз. М.: Мир, 1984.

P.V. Kovalev

### **GERT-BASED RESEARCH OF SOFTWARE BUILD ON N-VERSION ARCHITECTURE**

*In this article the technique of research the reliability of N-version software is offered. It allows using algorithms and methods of network analysis for researching the reliability of software, developed by using N-version approach.*

*Keywords: reliability, N-version, networks, estimation methods, network analysis.*