УДК 681.3

ОЦЕНКА НАДЕЖНОСТИ СЛОЖНЫХ ПРОГРАММНЫХ СИСТЕМ

Штарик А.В.
Научный руководитель - доцент Царев Р.Ю.
Сибирский федеральный университет

Целью проводимых исследований является оценка надежности сложных программных систем с помощью предложенной универсальной модели. В качестве сложной программной системы выступает объектно-ориентированное мультиверсионное программное обеспечение с распределенной архитектурой

Обозначения переменных, используемых в модели оценки надежности:

F - общее число компонентов (классов) в архитектуре программного обеспечения;

R_i - коэффициент надежности компонента i, i=1,…,F;

C_i - стоимость разработки компонента i, i=1,…,F;

Z_i - множество версий компонента i, i=1,…,F;

PU_i - вероятность использования компонента i, i=1,…,F;

PF_i - вероятность появления сбоя в компоненте i, i=1,…,F;

PL_{ij} - условная вероятность появления сбоя в компоненте i при сбое в компоненте j, i=1,…,F, j=1,…,F;

TA_i - относительное время доступа к компоненту i, i=1,…,F;

TC_i - относительное время анализа сбоя в компоненте i, i=1,…,F;

TE_i - относительное время устранения сбоя в компоненте i, i=1,…,F;

TU_i - относительное время использования компонента i, i=1,…,F;

TR - среднее время простоя системы;

MTTF - среднее время появления сбоя;

S - коэффициент готовности системы;

R_S - коэффициент надежности системы.

Для данной модели обязательно выполнение условия:

$$\sum_{i=1}^{F} PU_i = 1.$$

Надежность мультиверсионного компонента зависит от надежности каждой версии и от надежности мета-класса, реализующего механизм мультиверсионности:

$$R_i = (1 - \prod_{K \in Zi} PF_{ik}) R_{mul},$$

где R_{mul} – надежность мета-класса, реализующего механизм мультиверсионности. Этот класс не должен рассматриваться как компонент архитектуры и должен быть исключен из расчетов TR, MTTF и R_S.

Среднее время простоя системы вычисляется как:

$$TR = \sum_{i=1}^{F} [PU_i \times PF_i \times [(TA_i \times TC_i \times TE_i) + \sum_{j=1, j \neq i}^{F} [PL_{ji} [(TA_j \times TC_j \times TE_j)]]]].$$

Среднее время появления сбоя вычисляется как:

$$MTTF = \sum_{i=1}^{F} [PU_i \times (1 - PF_i) \times [TU_i + \sum_{j=1, j \neq i}^{F} [(1 - PL_{ji}) \times TU_j]]].$$

Коэффициент готовности системы вычисляется как:

$$S = MTTF/(MTTF + TR).$$

Коэффициент надежности системы вычисляется как:

$$R_S = \sum_{i=1}^{F} PU_i \times R_i \text{, где } R_i = 1 - \prod_{k \in Zi} PF_{ik}.$$

Для решения поставленной задачи исследования была разработана программа, позволяющая эмулировать поведение объектно-ориентированного мультиверсионного программного обеспечения с распределенной архитектурой.

Выявлено, что среднее время появления сбоя возрастает с возрастанием числа компонентов. Заметим, что самое маленькое время простоя не отражено для самого большого среднего времени появления сбоя. Причина заключается в том, что среднее появления сбоя зависит от времени использования компонентов, которое повышается для меньшего числа компонентов, но не обратно пропорционально числу компонентов.

Поскольку среднее время появления сбоя зависит от времени использования компонентов, то для эффективности компонентов, меньшее среднее появления сбоя не обязательно отражает более низкую надежность всей архитектуры.

В зависимости от количества и величины компонентов условные и безусловные вероятности сбоя, доступа, анализа и времени восстановления, а также и времени использования компонентов различны. Модель может использоваться, чтобы оценить надежность программного обеспечения для возможных архитектурных изменений и выбрать надежную архитектуру из различных вариантов. Однако модель не предлагает оптимальную надежную архитектуру программного обеспечения. Предполагается, что нахождение оптимальной архитектуры – это np-полная проблема.

Возможный выбор при разработке реальной системы ограничен многими факторами. Эти факторы включают аппаратную архитектуру, составляющие зависимости, которые не могут быть изменены, архитектурные уровни, которые не могут быть разделены или объединены, код компонента, который невозможно изменить, и другие ограничения.

Таким образом, подмножество возможных изменений архитектуры программного обеспечения в реальной среде обычно включает несколько вариантов. Модель можно применить ко многим вариантам для оценки надежности программного обеспечения при выборе лучшего варианта реализации.

Исследование позволило подтвердить применимость универсальной модели оценки параметров надежности программных средств как для оценки надежности мультиверсионного программного обеспечения, так и для традиционного программного обеспечения с распределенной архитектурой. А также позволило проверить применимость модели на разных этапах разработки современного программного обеспечения.

Исследование выявило особенности предложенной универсальной модели оценки надежности объектно-ориентированного мультиверсионного программного обеспечения с распределенной архитектурой, а также позволило определить область ее применения и надежность результатов, получаемых на разных этапах разработки программного обеспечения с различной архитектурой.