

## ТЕСТИРОВАНИЕ И ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ушаков Д.Е.

В статье рассмотрено тестирование и верификация программного обеспечения для ввода зданий и расчётного модуля данной программы на примере программы PlanMaker.

Задача тестирования и верификации программных систем занимают центральное положение в исследованиях по математической теории программирования. Это обусловлено, в первую очередь, высокой актуальностью создания теоретического фундамента для разработки надёжного программного обеспечения. Современный этап развития индустрии программных систем характеризуется значительным усложнением процесса их разработки.

Нашей задачей было выявление проблем тестирования и верификации на примере отладки программного обеспечения для расчета риска при пожарной эвакуации [1,2,3]. Основная проблема состояла в разработке тестов, позволяющих судить о качестве программы.

Требования качества отражены в стандарте [5], наиболее важные из них:

- Корректность, т.е. соответствие системы предназначению;
- Безопасность, отсутствие неавторизованной утечки информации;
- Устойчивость системы в случае непредусмотренного поведения окружения и при работе с неправильными входными данными;
- Эффективность использования ресурсов времени и памяти;
- Оптимальность реализованных в системе алгоритмов.

Как правило, анализ соответствия предъявленным требованиям проводится либо путём визуального анализа, либо методом тестирования. Если какое-либо из свойств системы может быть выражено формально, например, в виде математической формулы, то анализ этого свойства может быть проведён методами верификации. Так эти методы были использованы для отладки и проверки программного обеспечения процесса эвакуации при пожаре.

Тестирование программы заключается в анализе результатов на некоторых выборочных данных. Тестирование является в настоящее время основной формой контроля качества программ, и занимает большую часть разработки самой программы. Такими тестами являлись тесты на нагрузку расчётов, путём увеличения расчётных данных и сокращения времени расчётов, и тестирования выходных данных в соответствии с теоретическими расчётами.

Однако, тестирование обладает фундаментальным недостатком: если его невозможно провести для всех входных данных, а только лишь для небольшой их части, то оно не может служить гарантированным обоснованием того, что программа обладает проверяемыми свойствами.

Таким образом, заострим своё внимание на верификации.

Верификация состоит из следующих частей:

- Построение математической модели анализируемой системы;
- Представление проверяемых свойств в виде формального текста (называемого спецификацией);
- Построение формального доказательства наличия или отсутствия у данной программы проверяемых свойств. Математическое моделирование представляет собой граф, где вершинами являются состояния, и изображают ситуации, в которых может находиться программа в разные моменты времени, и рёбра которого могут иметь метки, изображающие действия.

Спецификация – это описание свойств системы в виде формального текста, которая может выражать связь между вводными значениями или зависимость между свойствами системы. Построение доказательств, где существует два основных метода построения доказательств того, что модель соответствует спецификации [4]:

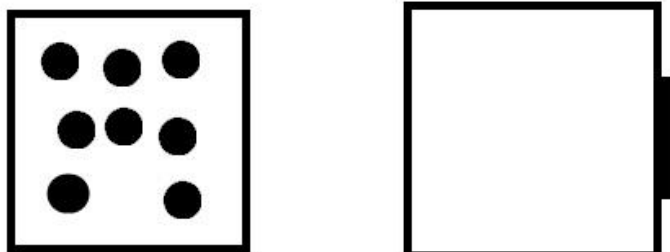
- Model checking;

-Логический вывод.

В нашем случае разработана и проведена серия верификационных тестов для проверки некоторых свойств программы.

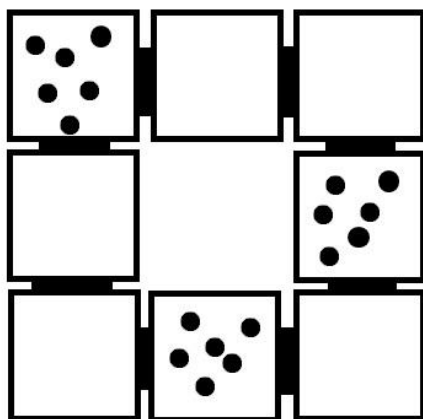
Краш – тесты, основная задача которых выявить явное нарушение работы программы в некоторых сценариях действий пользователя и их предотвращение.

Тест №1 (рис.1) позволяет определить наличие в программе механизмов контроля связности и целостности здания, т.е. наличие путей эвакуации.



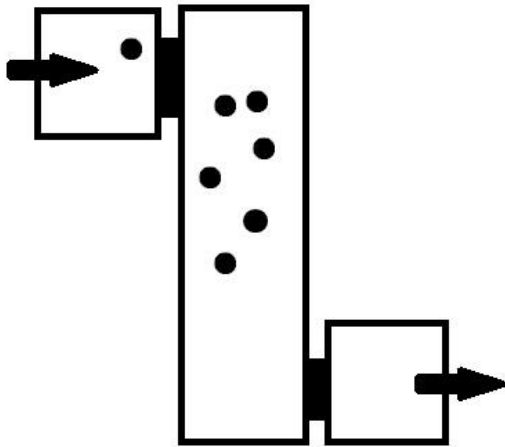
**Рис.1** Тест №1

Тест №2 (рис.2) определяет завершенность введения здания, присутствия в нём выходов для эвакуации и отсутствие свойства ведущего к закливлению процессов расчёта пожарного риска.



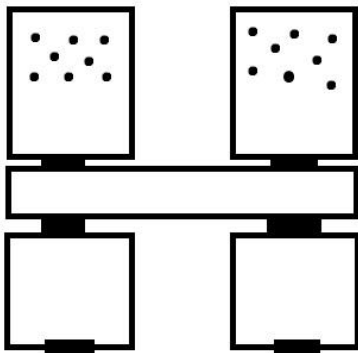
**Рис.2** Тест №2

Тест №3 (рис.3) осуществляет проверку отсутствия закливления путей эвакуации, т.е. попадание людей в пройденные участки пути, возврат в прежние комнаты и схожие ситуации.

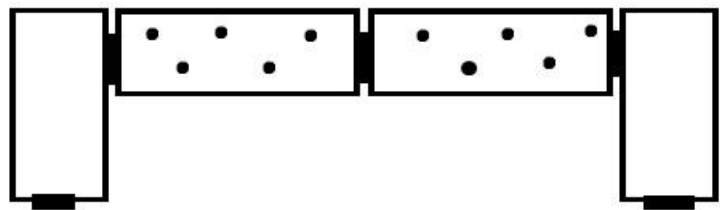


**Рис.3** Тест №3

Симметрические тесты, позволяющие определить равномерность и правильность вычисления симметричных ситуаций, такие как тесты №№4-5 (рис.4,5) проверка симметричности вычисления.

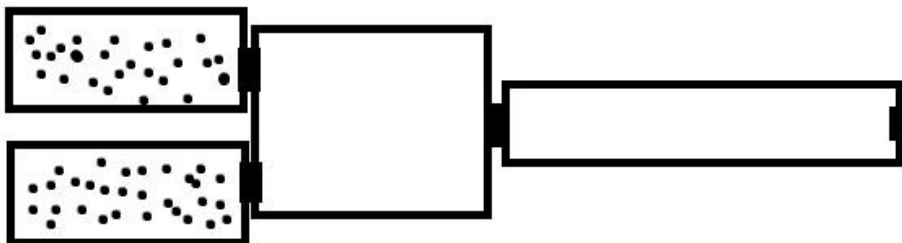


**Рис.4** Тест №4



**Рис.5** Тест №5

Тесты №№6-7 (рис.6,7) на заторы, позволяющие определить соответствие теоретической постановке задачи (ГОСТ эвакуации) с вычислительным модулем, определение ситуации и характерных свойств заторов.



**Рис.6** Тест №6

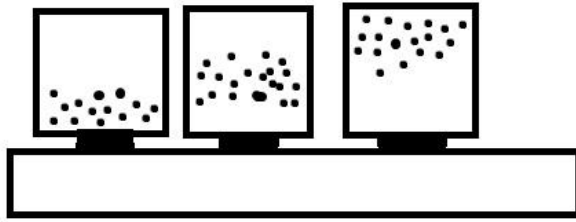


Рис.7 Тест №7

Впоследствии все тесты использовались в тестировании нагрузки на вычислительный модуль путём увеличения количества людей в комнате и изменения путей и количества комнат.

	50 человек	100 человек	1000 человек	10000 человек	100000 человек
Тест 4	0,0001мс	0,0002мс	0,0023мс	0,0194мс	0,1913мс
Тест 5	0,0001мс	0,0002мс	0,0021мс	0,0191мс	0,1916мс
Тест 6	0,0006мс	0,0012мс	0,0128мс	0,0191мс	—*
Тест 7	0,0009мс	0,0018мс	0,0197мс	—*	—*

— \* ситуации, когда происходят заторы.

Таким образом, проведя серию экспериментов тестирования и верификации программного обеспечения процесса эвакуации при пожаре, необходимо отметить, что найдены линейные зависимости нагрузки на расчётный модуль программы, что ведёт к линейным затратам времени для оценки риска, найдены характерные свойства для закливания расчётов, их полной остановки. Необходимые данные позволяют сделать выводы о недоработке инструментов ввода здания для расчёта эвакуации, дают статистику для оптимизации расчётного модуля и нового инструмента ввода здания, исключающего элемент человеческого фактора, а, именно, обучаемость, отсутствие навыков работы с персональными компьютерами. Всё это позволяет сделать вывод о необходимости и важности продолжения тестирования данного продукта и других продуктов в следующих разработках.

### Список литературы

1. Морозов, О. А. Архитектура, методология создания и принципы работы сервера "Безопасность в техносфере" // Безопасность в техносфере : сб. статей / Удмурт. регион. отд-ние Общерос. обществ. орг. "Рос. науч. о-во анализа риска", ГОУВПО "Удмурт. гос. ун-т", Учеб.-науч. ин-т природ. и техноген. катастроф ; науч. ред.: В. М. Колодкин, И.Л. Бухарина. - Ижевск : Удмурт. ун-т, 2010. - Вып. 6. - С. 35-42.
2. Морозов, О. А. Открытая платформа интеграции сервисов "Безопасность в техносфере" // Производство. Технология. Экология : междунар. конф. с элементами науч. шк. для молодежи : материалы конф. и шк. / Гл. упр. МЧС РФ по УР, ГОУВПО "Удмурт. гос. ун-т", Ин-т исслед. природ. и техноген. катастроф. - Ижевск, 2010. - С. 20-24. - Библиогр.: с. 23-24 (7 назв.).
3. Проблемно-ориентированный сервис "Декларация пожарной безопасности общественных зданий и сооружений" // Безопасность в техносфере : сб. статей / Удмурт. регион. отд-ние Общерос. // обществ. орг. "Рос. науч. о-во анализа риска", ГОУВПО "Удмурт. Гос. // ун-т", Учеб.-науч. ин-т природ. и техноген. катастроф ; науч. ред.: В. М. // Колодкин, И.Л. Бухарина. - Ижевск : Удмурт. ун-т, 2010. - Вып. 6. - С. 11-21. - Библиогр.: с. 21 (7 назв.).

4. Миронов А. М., Жуков Д. Ю. Математическая модель и методы верификации программных систем // Интеллектуальные системы. Т. 9. 2005. Вып. 1–4, с. 209–252
5. International Standard ISO/IEC 9126.// Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for their Use.// International Organization for Standardization, International Electro technical Commission, Geneva, 1991.