

ИССЛЕДОВАНИЕ НЕОБХОДИМОСТИ ИСПОЛЬЗОВАНИЯ ВЕРИФИКАЦИИ ПРОГРАММ

Нещадым В.А., Щевцова О.С.

ГВУЗ «Национальный горный университет», <http://www.nmu.org.ua>

Обеспечение высокой надежности и безошибочности программ — одна из главных задач индустрии информационных технологий. Использование верификации, дополняющей традиционные методы тестирования и отладки, позволяет повышать качество программ.

Ключевые слова – верификация программ; ПО; метод формальной верификации.

ВСТУПЛЕНИЕ

Сложность программ давно подошла к границе их понимания человеком, а следовательно, к границе их управляемости, поэтому постоянно растет число ошибок в разработанных и сданных заказчику программных системах. Особенно подвержены ошибкам параллельные, распределенные и многопоточные программы, характерные для систем управления, причем именно такие программы получают сейчас все большее распространение - например, встраиваемые системы управления в автомобильной и авиационной промышленности, очень критичные к требованиям безопасности [1].

ОСНОВНАЯ ЧАСТЬ

Верификацией называется проверка соответствия результатов отдельных этапов разработки программной системы требованиям и ограничениям, сформулированным для них на предыдущих этапах. Верификация проверяет соответствие одних создаваемых в ходе разработки и сопровождения ПО артефактов другим, ранее созданным или используемым в качестве исходных данных, а также соответствие этих артефактов и процессов их разработки правилам и стандартам. В частности, верификация проверяет соответствие между нормами стандартов, описанием требований (техническим заданием) к ПО, проектными решениями, исходным кодом, пользовательской документацией и функционированием самого ПО. Кроме того, проверяется, что требования, проектные решения, документация и код оформлены в соответствии с нормами и стандартами, принятыми в данной стране, отрасли и организации при разработке ПО, а также — что при их создании выполнялись все указанные в стандартах операции, в нужной последовательности. Обнаруживаемые при верификации ошибки и дефекты являются расхождениями или противоречиями между несколькими из перечисленных документов, между документами и реальной работой программы, между нормами стандартов и реальным процессами разработки и сопровождения ПО. При этом принятие решения о

том, какой именно документ подлежит исправлению (может быть, и оба) является отдельной задачей [2].

Сегодня в области верификации произошли глубокие изменения — был разработан метод формальной верификации model checking (проверка модели), который стал теоретической основой эффективных алгоритмов, позволяющих проверять правильность промышленных программно-аппаратных систем [3].

Задачами верификации в рамках жизненного цикла программного обеспечения являются:

- Выявление наиболее критичных и наиболее подверженных ошибкам частей создаваемой или сопровождаемой системы;
- Выявление дефектов (ошибок, недоработок, неполноты и пр.) различных артефактов разработки ПО (требований, проектных решений, документации или кода), что позволяет устранять их и поставлять пользователям и заказчикам более правильное и надежное ПО;
- Контроль и оценка качества ПО во всех его аспектах;
- Предоставление всем заинтересованным лицам (руководителям, заказчикам, пользователям и пр.) информации о текущем состоянии проекта и характеристиках его результатов [2].

ВЫВОД

При построении систем определенного уровня сложности люди в принципе не могут избежать ошибок, просто потому, что им вообще свойственно ошибаться, а возрастающая сложность предоставляет все больше возможностей для ошибок, при этом затрудняя их быстрое обнаружение. Для обеспечения корректности и надежности работы таких систем большое значение имеют различные методы верификации [1].

Сегодня верификация становится одним из основных методов достижения высокой надежности программного обеспечения в дополнение к традиционным методам тестирования и отладки. В последние несколько лет в области верификации произошли существенные изменения, превратившие ее из «игры разума» теоретиков в практически методики анализа реальных программ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. В.В. Кулямин «Методы верификации программного обеспечения».
2. Юрий Карпов «Новая жизнь верификации»
3. Карпов Ю. Г. Model checking. Верификация параллельных и распределенных программных систем. // БХВ Петербург, 2010, 560 с.