

Предисловие

Очередной сборник трудов Института системного программирования почти полностью (но не совсем) посвящен проблематике тестирования компьютерных программных и аппаратных средств.

В статье В. В. Кулямина «Организация сложных тестовых наборов» рассматриваются различные способы организации и структуризации сложных тестовых наборов. Анализируются основные проблемы эксплуатации и развития тестов, для решения которых необходимо введение дополнительной структуры. Рассматриваются три базовых техники структуризации тестов — выделение модулей, определение квалификаторов и введение конфигурационных параметров.

В статье Д. В. Силакова и А. В. Хорошилова «Методы обеспечения переносимости ПО» обсуждается проблема переносимости приложений между программно-аппаратными платформами. Предлагается обзор подходов к решению этой проблеме, появившихся за время развития ИТ, а также анализ преимуществ и недостатков каждого из них. Рассматриваются области применения существующих решений.

Статья Н. В. Пакулина и С. А. Смолова «Применение технологии UniTESK для функционального тестирования инфраструктурного ПО Грид» посвящена вопросам тестирования для инфраструктурного программного обеспечения (ИПО, middleware) Грид-систем на соответствие стандарту. В статье рассматривается разработка тестовых наборов для ИПО Грид средствами технологии автоматизированного тестирования UniTESK. Приведены результаты тестирования программного пакет Globus Toolkit 4.2 на соответствие базовому стандарту WSRF 1.2.

П. Н. Яковенко и А. В. Сапожников представили статью «Инфраструктура тестирования веб-сервисов на базе технологии TTCN-3 и платформы .NET», в которой обсуждаются вопросы тестирования веб-сервисов с использованием технологии TTCN-3. Рассматривается отображение языка WSDL в TTCN-3, основанное на синхронном (процедурном) взаимодействии тестового сценария с веб-сервисом. Предлагается подход к реализации универсального тестового адаптера средствами рефлексии, предоставляемыми платформой .NET, и средой поддержки времени выполнения языка TTCN-3.

В статье А. В. Баранцева, С. В. Грошева и В. А. Омельченко «Генерация оптимизированных для ручного выполнения сценариев тестирования приложений с графическим интерфейсом пользователя» описывается решение задачи построения последовательностей действий пользователя, оптимизированных для ручного выполнения, на основе модели в виде диаграммы состояний и переходов.

В статье М. В. Архиповой и С. В. Зеленова «Направленная генерация тестовых данных для анализаторов статической семантики» представлен метод

SemaTESK автоматической генерации множеств тестов для фронт-эндов в трансляторах. Метод ориентирован на тестирование анализаторов статической семантики. Предложенный метод специфицирования статической семантики позволяет формализовать неформальные требования, содержащиеся в нормативных документах (например, в стандартах). Метод SemaTESK был успешно применен в ряде проектов, в том числе по тестированию анализаторов статической семантики языков C и Java.

В статье Д. Н. Воробьева и А. С. Камкина «Генерация тестовых программ для подсистемы управления памятью микропроцессора» рассматривается методика генерации тестовых программ, предназначенных для функционального тестирования подсистемы управления памятью микропроцессора. Предлагаемая методика основана на формальной спецификации инструкций, работающих с памятью, и устройств подсистемы, включая буфер трансляции адресов и кэш-память. Для генерации тестовых программ в предлагаемом подходе используются комбинаторные техники — тестовые воздействия на микропроцессор представляются в виде цепочек инструкций небольшой длины, составленных из различных сочетаний тестовых ситуаций.

Статья Я. С. Губенко, А. С. Камкина и М. М. Чупилко «Сравнительный анализ современных технологий разработки тестов для моделей аппаратного обеспечения» посвящена сравнению современных подходов к разработке функциональных тестов для моделей аппаратуры: AVM (Advanced Verification Methodology) от компании Mentor Graphics, OVM (Open Verification Methodology) — совместной разработки Mentor Graphics и Cadence Design Systems — и технологии UniTESK (Unified TEsting and Specification tool Kit), разработанной в Институте системного программирования РАН.

В статье Е. В. Корныхина «Генерация тестовых данных для системного функционального тестирования микропроцессоров с учетом кэширования и трансляции адресов» рассматривается задача генерации тестовых данных для системного функционального тестирования микропроцессоров (core-level verification), а именно задача построения тестовой программы по заданной ее абстрактной форме (тестовому шаблону). Для решения этой задачи в работе предложен алгоритм, сводящий ее к задаче разрешения ограничений.

Статья И. Б. Бурдонова и А. С. Косачева «Полное тестирование с открытым состоянием ограниченно недетерминированных систем» посвящена описанию подхода к проблеме полноты тестирования, под которым понимается проверка соответствия реализации требованиям, описываемым спецификацией. Тестирование полное, если оно обнаруживает все возможные ошибки в реализации. Для практического применения тестирование должно заканчиваться за конечное время. Требования полноты и конечности тестирования в общем случае взаимно противоречат друг другу. Однако для ограниченных классов реализаций и спецификаций, а также при использовании дополнительных тестовых возможностей, удаётся построить

конечные полные тесты. Предлагаются алгоритмы тестирования и даётся оценка их сложности для конечных спецификаций и конечных реализаций с ограниченным недетерминизмом при тестировании с открытым состоянием.

Наконец, в статье тех же авторов «Тестирование с преобразованием семантик» представлен подход к формальному описанию тестирования в случае, когда тестовые семантики спецификации и реализации различаются. Для этого используется медиатор, осуществляющий преобразование спецификационных тестовых воздействий в реализационные и обратное преобразование реализационных наблюдений в спецификационные. Формально переопределяются условия тестирования и конформность. Также рассматривается тестирование с заданным преобразованием реализационных состояний в спецификационные. Определяются условия и способ построения теста, обнаруживающего все ошибки в реализации за конечное время.

Академик РАН В.П. Иванников