

Е. БРИТВИНА, Н.Л. КАЗАКОВА<sup>1</sup>, В.В. КУЛЯМИН,  
А.К. ПЕТРЕНКО

*Институт системного программирования РАН, Москва*

*<sup>1</sup>ОО "диФэктори Сарл"*

## **UNITESK – ТЕХНОЛОГИЯ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Рассматриваются вопросы внедрения и использования современной технологии тестирования ПО на основе формальных спецификаций.

Технология и набор инструментов UniTesK предназначен для поддержки функционального тестирования UniTesK позволяет формализовать требования к ПО, строго определять критерии полноты тестирования, автоматически создавать тесты на основе требований и критериев тестирования, формировать и поддерживать отчетность по процессу тестирования.

Сфера применения инструментов UniTesK не ограничивается некоторой узкой предметной областью, это система общего назначения. Гибкая архитектура тестов UniTesK позволяет свести к минимуму работу по адаптации тестовой системы к конкретному приложению. UniTesK нацелен на использование в реальных производственных процессах и помогает решить две главные задачи: повысить качество ПО и при этом сделать процесс тестирования экономичным. Качество достигается за счет того, что набор тестов строится по четко сформулированной схеме, и тестирующим волею неволей приходится проверять разные, часто "неразумные" ситуации, а именно в них наличие ошибок наиболее вероятно. Экономичность достигается четким разделением функциональных частей тестовой системы. Опыт применения UniTesK показывает, что такое разделение существенно удешевляет поддержку тестовых наборов по ходу эволюционного развития целевых систем, удешевляя масштабируемость, сопровождаемость и переиспользование тестовой системы.

Технология UniTesK максимально приспособлена для внедрения по методу от простого к сложному (от минимальных затрат к промышленным). В начале процесса внедрения рекомендуется охватить тестами только часть компонент и при этом описать требования не подробно. Этой информации будет достаточно, чтобы сгенерировать автоматически тестовые последовательности, выполнить их и оценить

полученное качество тестирования. Затем для тех компонентов и интерфейсов, которые требуют повышенного внимания, можно провести уточнение спецификаций поведения и задать параметры генерации тестов, которые позволяют повысить качество тестирования (если качество мерить в процентах покрытия исходного кода, то реально достигать 80-90% даже на крупных приложениях). Тем самым, процесс позволяет, с одной стороны, достигать высоких показателей качества, с другой стороны, процесс достаточно гибок и не требует тратить одинаковые усилия как на критические компоненты, так и на относительно неважные компоненты.

UniTesK предполагает, что тестированием занимаются люди с хорошим опытом программирования и проектирования.

Инструмент Ch@se технологии UniTesK был успешно применен для платформы .NET для тестирования Windows-сервисов. Windows-сервисы - приложения без пользовательского интерфейса, в промышленных информационных системах обычно предназначены для обработки данных на периодической основе, без участия пользователя. В качестве обрабатываемых данных могут быть файловая система, базы данных, xml-файлы и т.п. Применение технологии UniTesK при тестировании Windows-сервисов, работающих с различными типами данных, в целом оказалось успешным. Основные особенности применения заключались в таких компонентах UniTesK как медиаторы и сценарии. В медиаторах, задающих соответствие интерфейсов модели и объекта, пришлось искать "обходные" пути для вызова метода объекта в случае, если он не предоставляет к ним доступа. В сценариях же основная сложность состояла в моделировании состояния и переходов из одного состояния в другое.

#### *Список литературы*

1. <http://www.ispras.ru/groups/rv.html>
2. Бритвина Е., Грошев С., Монахов А., Петренко А., Петренко О. Тестирование на основе моделей. // ОТКРЫТЫЕ СИСТЕМЫ, №9, 2003
3. <http://www.citforum.ru/SE/testing/model>
4. Кулямин В.В., Омельченко В.А., Петренко О.Л. Обучение передовым технологиям разработки ПО: проблемы и методы их решения // Труды ИСП РАН, №5, 2004, с. 101-120
5. Кулямин В. В., Петренко А. К., Косачев А. С., Бурдонов И. Б. Подход UniTesK к разработке тестов. // ПРОГРАММИРОВАНИЕ, 29(6), 2003
6. V.V.Kuliamin, A.K.Petrenko. Theory and Practice of Co-verification Process. UniTesK Story// ETAPS'2003 Tutorial, Warsaw, April 2003. Bourdonov, A. Kossatchev, V. Kuli Amin, A. Petrenko. UniTesK Test Suite Architecture. Proceedings of FME 2002. LNCS 2391, pp. 77-88, Springer-Verlag, 2002.