

УДК 004.054

Б.В. Тютин, А.О. Веселов, В.П. Котляров

ТЕСТИРОВАНИЕ НА ОСНОВЕ КЛЮЧЕВЫХ СЛОВ С ИСПОЛЬЗОВАНИЕМ ДИАГРАММ ПОСЛЕДОВАТЕЛЬНОСТИ СОБЫТИЙ

B.V. Tiutin, A.O. Veselov, V.P. Kotlyarov

SEQUENCE DIAGRAMS APPLICATION FOR KEYWORD-DRIVEN TESTING

Рассмотрен подход к тестированию на основе ключевых слов, базирующийся на тестовых сценариях в формате диаграмм Message Sequence Charts. Изучены основные особенности и преимущества данного подхода. Приведено краткое описание существующего программного решения по автоматизации тестирования, в основе которого заложены представленные идеи.

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ; ТЕСТИРОВАНИЕ НА ОСНОВЕ КЛЮЧЕВЫХ СЛОВ; ФОРМАЛЬНЫЕ СПЕЦИФИКАЦИИ; MESSAGE SEQUENCE CHARTS.

The given paper focuses on the application of sequence diagrams in Message Sequence Chart format for keyword-driven testing. We have studied a keyword-driven approach and presented main advantages of sequence diagrams. The paper comes up with a brief description how to implement testing automation framework based on presented techniques.

TESTING AUTOMATION; KEYWORD-DRIVEN TESTING; FORMAL SPECIFICATION; MESSAGE SEQUENCE CHARTS.

Основным требованием к промышленным технологиям производства программного продукта (ПП) является возможность построения единой автоматизированной технологической цепочки, обеспечивающей в том числе контроль качества разрабатываемых продуктов. Поэтому в текущих условиях наибольшую эффективность обеспечивают процессы, базирующиеся на инструментальных средствах, функциональность которых может быть легко адаптирована или расширена к условиям бесшовной интеграции в рамках технологической цепочки.

Настоящая статья сфокусирована на особенности реализации эффективной технологической цепочки тестирования.

Реализация инструментов тестирования на основе известного подхода, управляемого данными (Data-Driven Testing – DDT) [1], позволяет решить проблемы автоматизации, снижения сложности и обеспечения понимания широким кругом специалистов. Среди разнообразия существующих подходов к управлению тестированием на основе

данных, наиболее перспективно тестирование на основе ключевых слов (Keyword-Driven Testing – KDT) [2]. Этот подход упрощает разработку и поддержку тестового набора путем отделения логики тестовых процедур от их реализации.

Разработка гибких и в то же время интегрированных в единую технологическую цепочку инструментов тестирования является актуальной задачей, решение которой позволяет снизить затраты на поиск дефектов в программном обеспечении. В данной статье рассматривается подход к автоматизации тестирования, объединяющий дружественную для пользователя нотацию MSC (Message Sequence Charts) [3] и метод тестирования на основе ключевых слов.

Подход, основанный на ключевых словах

Тестирование, основанное на ключевых словах, позволяет создавать и выполнять структурированные сценарии, в которых данные отделены от потока управления.

Сценарий теста состоит из списка ключевых слов и их параметров. Каждое ключевое

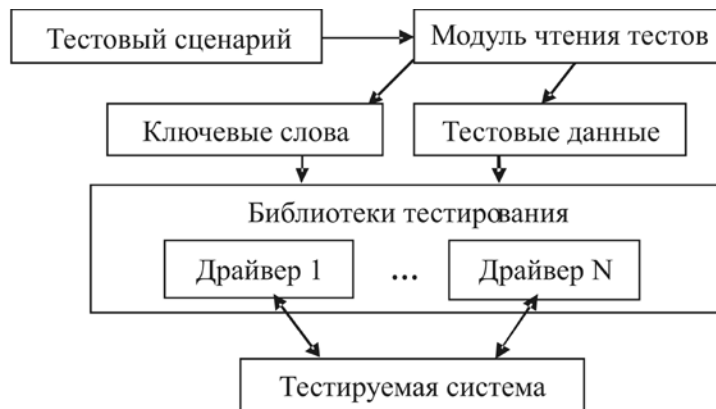


Рис. 1. Схема тестирования на основе ключевых слов

ключевое слово представляет собой заданный перечень воздействий, выполняемых над тестируемой системой. Сам сценарий имеет табличный формат и может быть отредактирован как в обычном текстовом редакторе, так и при помощи специальных программных инструментов. Рис. 1 иллюстрирует концепцию подхода к тестированию на основе ключевых слов в случае автоматизированного тестирования.

Каждое ключевое слово имеет однозначное определение, которое может быть задано в виде кода для автоматизированного тестирования или в виде текстовой процедуры для ручного тестирования. В первом случае при помощи специального модуля осуществляется чтение и разбор тестового сценария. Найденные ключевые слова и их аргументы интерпретируются, после чего происходит выполнение соответствующих им действий. Программный код, реализующий данные действия, компонуется в библиотеки тестирования, также называемые драйверами. Именно они осуществляют взаимодействие с тестируемой системой и проводят регистрацию событий.

При таком подходе возможно создание определений ключевых слов отдельно и независимо от создания тестовых сценариев. Таким образом, создание тестов разделено на две независимых активности: разработка тестовых сценариев и создание тестовых действий – элементов, из которых строятся сценарии (ключевых слов).

При использовании подхода, основан-

ного на ключевых словах, затраты на создание и сопровождение тестового набора в среднем снижаются на 25 %. Это обусловлено тем, что изменения требований, в основном, требуют не изменения сценариев, а только уточнения определений ключевых слов. Тестирование становится масштабируемым, что означает возможность распределенного выполнения набора тестов среди имеющихся вычислительных ресурсов [4]. Модификация тестовых сценариев упрощается тем, что тесты оперируют высокоуровневыми абстракциями и переиспользуют существующие ключевые слова, что в свою очередь снижает требования к компетенции тестировщика в детальном знании объекта тестирования.

Преимущества MSC для тестирования

MSC описывают асинхронное взаимодействие нескольких процессов. Проще говоря, они представляют собой описание набора инстанций, которые взаимодействуют друг с другом при помощи обмена сигналами [3].

На практике оказывается, что использование диаграмм последовательности событий (MSC) является очень удобным для представления сценариев, для описания конкретных спецификаций телекоммуникационных протоколов или для протоколов обмена данными между различными компонентами разрабатываемой системы. Текстовое представление MSC легко может быть разобрано и проанализировано. Все это по-

зволяет использовать MSC для описания тестовых сценариев. В настоящей работе замечено, что концепция сигнала MSC очень близка к концепции ключевого слова. Оба понятия содержат информацию о тестовом действии и наборе данных для него, имеют контекстно-независимую семантику и могут быть однозначно интерпретированы согласно установленной процедуре тестирования.

MSC как формат теста, основанного на ключевых словах, имеет следующие преимущества. Он может быть представлен в графическом виде [5] и легко воспринимается человеком. Так как формат стандартизирован, возможно переиспользование сторонних инструментов и библиотек для разбора MSC. Некоторые существующие инструменты моделирования и разработки позволяют выдавать данные в формате MSC как для документирования поведения системы, так и для генерации тестов.

Сравнение возможностей существующих технологий позволило выявить ключевые аспекты функциональности, требуемые для автоматизации тестирования, а также существующие пробелы (см. табл.). Были рассмотрены системы и фреймворки тестирования Fit [6], Robot [7], JUnit [8], UniTesK [9], TAT [10], HP Quick Test Professional (QTP) [11], IBM Rational Functional Tester (RFT) [12].

Базовым элементом рассмотренных систем тестирования является хотя бы частичная реализация подходов DDT и KDT. Это позволяет создавать параметризованные тесты, в которых данные отделены от логики тестирования. Вторым по важности аспектом является возможность создания в рамках применяемого формата тестов

структурированных сценариев. Это позволяет разделить процессы по созданию и расширению тестового набора, задействовав в них различных специалистов.

Редактирование тестов в графическом представлении является менее трудоёмким процессом, в котором может быть задействован менее квалифицированный пользователь. Возможность графического представления тестовых сценариев имеется у всех технологий, за исключением JUnit и QTP, однако возможность их редактирования доступна только у TAT и RFT.

Важным аспектом языка тестовых сценариев является возможность реализовывать сложную, нелинейную логику, описывающую альтернативные ветки поведения, циклы, параллельные взаимодействия.

С точки зрения автоматизации тестирования рассмотренные инструменты предоставляют примерно равные возможности. Однако стоит отдельно отметить интеграцию JUnit в процесс разработки кода программного продукта, нашедшую широкое применение в Test Driven Development.

Использование web-технологий делает процесс формирования и визуализации результатов тестирования независимым от среды исполнения, что может использоваться для реализации облачного тестирования.

Концепция KDT на базе MSC

Основными элементами MSC диаграмм являются сигналы, действия (MSC actions) и нелинейные конструкции. Все они могут использоваться для реализации концепции тестирования, основанной на ключевых словах.

Сравнение каркасов построения систем тестирования

	Fit	Robot	JUnit	TAT	UniTesK	QTP	RFT
DDT	+	+/-	+/-	+/-	+/-	+/-	+
KDT	-	+	-	+/-	+/-	+/-	-
Параметризация тестов	-	+	-	+	+	+/-	+
Нелинейные сценарии	-	+	+	+	+	+	+
Параллельные сценарии	-	-	+/-	+	+/-	-	+/-
Отображение результатов тестирования в интерактивной форме	-	+/-	+	+	+/-	+	+

Обмен сигналами может интерпретироваться как последовательность ключевых слов, следующих одно за другим. Данные передаются как параметры сигналов. В контексте тестирования все экземпляры MSC диаграммы разделяются на две группы: принадлежащие к тестовому окружению и принадлежащие к тестируемой системе. Таким образом, все сигналы от окружения к системе интерпретируются как воздействия на систему, в то время как сигналы в обратном направлении интерпретируются как ответы системы на воздействия от окружения. В тестировании, основанном на ключевых словах, различия в направлении сигналов стираются, т. к. каждый сигнал превращается в команду — ключевое слово. Таким образом, концепцию посылки и приема сигналов MSC становится необходимо адаптировать для ключевых слов таким же образом, чтобы сделать тесты читаемыми.

Для этого предложено рассматривать сигналы от окружения к системе как ключевые слова, непосредственно отвечающие за действия над системой, а сигналы от системы к окружению — как ключевые слова, отвечающие за проверку ответов системы, либо ее состояний.

Действия (actions) в MSC используются для описания внутренних событий экземпляров, таких как вычисление некоторых значений, вызов функций, добавление комментария. Как элемент тестового сценария, действия нередко используются для повышения читаемости теста за счет указания в них комментариев, кодирующих соответствующую активность. При ручном тестировании в действиях могут содержаться дополнительные инструкции или находиться ссылки на необходимую документацию или раздел требований. В автоматическом тестировании действия могут использоваться для вставки целевого кода в тест, в т. ч. и для передачи в тест ключевых слов.

Таким образом, оказывается возможным использование MSC для тестирования, основанного на ключевых словах, и использование компонент диаграмм для того, чтобы сделать тесты легко воспринимаемыми. Роль каждого элемента должна быть ясно определена в том случае, когда

от тестовых сценариев требуется восприятие человеком.

Нелинейные выражения MSC позволяют задавать различные нелинейные сценарии тестов, в частности возможно использование циклов (loop expression), опциональных (opt expression) или альтернативных (alt expression) поведений. В сочетании с условными конструкциями (MSC condition), нелинейные выражения позволяют использовать if-else конструкции и циклы как в обычных языках программирования. Переменные, используемые в действиях, параметрах сигналов и в выражениях условных конструкций, могут быть определены при помощи специальной конструкции text.

Создаваемые тестовые сценарии за счет предложенных в работе идей по интерпретации элементов MSC позволяют описывать взаимодействие распределенных компонент системы. Помимо этого, от существующих решений, представленных в работах по использованию MSC для тестирования, разработанный подход отличается тем, что представленные на диаграмме недетерминированные варианты поведения могут определять не только взаимоисключающие, но и равновероятные варианты.

Следует отметить, что MSC как стандартизованная нотация позволяет с легкостью создавать на ее основе тесты, основанные на ключевых словах. Система тестирования по ключевым словам, в свою очередь, может автоматически запускать тесты, используя различные нижележащие драйверы и библиотеки.

В рамках данной работы была создана автоматизация системы тестирования, включающая систему трансляции тестовых сценариев с MSC представления в исполняемое представление, основанное на ключевых словах.

Реализация системы тестирования

Основная идея, реализованная в системе тестирования, заключается в вычлениении ключевых слов из MSC и их дальнейшем запуске при помощи драйверов и библиотек. Но вместо прямой интерпретации в команды в системе реализована



Рис. 2. Схема автоматизации тестирования на основе MSC

трансляция в целевой код. Такой подход нашел своё отражение в ряде работе по использованию MSC [13]. Полученный код теста на целевом языке представляет собой созданный по определенным правилам автомат, реализующий набор состояний и систему переходов, отражающих сценарий теста. Такая архитектура позволяет реализовать нелинейные конструкции MSC. Основные компоненты созданного программного комплекса проиллюстрированы на рис. 2.

В тестовых сценариях реализована возможность использования переменных,

которые могут применяться для задания тестовых данных и управления тестированием. Каждая инстанция MSC, относящаяся к тестовому окружению, выполняется в отдельном потоке и независима от остальных. Это требует наличия двух типов переменных: локальных (доступность которых ограничена рамками данной инстанции) и глобальных (которые доступны из любой инстанции). Для реализации безопасного доступа к глобальным переменным вводится дополнительный поток, отвечающий за хранение и доступ к глобальным переменным [14]. Данный вспомогательный

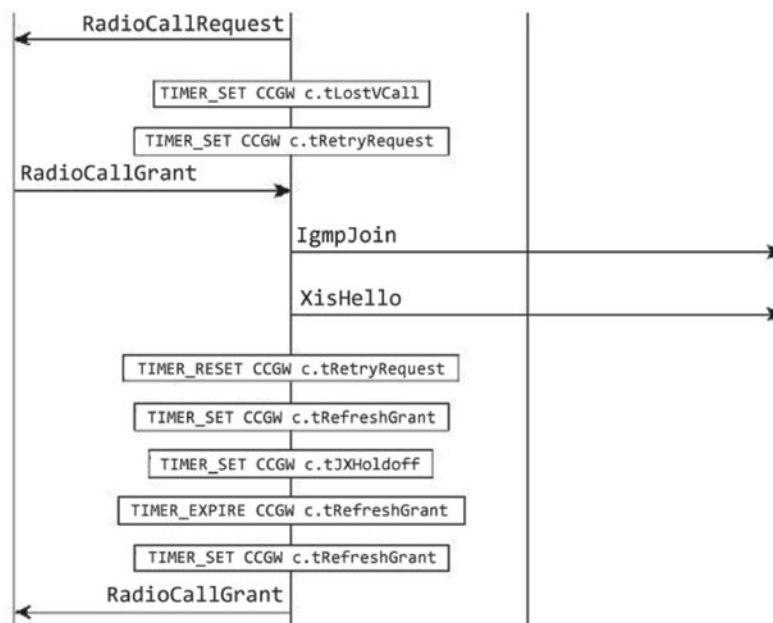


Рис. 3. Пример тестового сценария

поток также отвечает за принятие решений при выборе ветки альтернативы, когда решение зависит от нескольких инстанций сразу.

На рис. 3 приведен пример фрагмента тестового сценария в формате MSC для организации распределённого тестирования. Данный тест применялся в проекте по тестированию компонента встроенной телекоммуникационной системы, реализующей канал агрегации и передачи сигнальной информации в сети. Сценарий проверял взаимодействие модуля-медиатора с поставщиками информации.

На диаграмме представлены четыре взаимодействующих инстанции, одна из которых является тестируемой системой. Три оставшиеся эмулируют рабочее окружение, с которым тестируемая система обменивается сигналами (например, RadioCallGrant). Для иллюстрации ожидаемого изменения внутреннего состояния тестируемой системы использованы actions, содержащие псевдокод внутренних вызовов (например, TIMER_SET).

Представленное исследование ещё не завершено. В его рамках предложен подход реализации тестирования на основе ключевых слов с представлением тестовых сценариев в виде диаграмм последовательности событий и сформулированы правила интерпретации элементов MSC, обеспечивающие реализацию KDT.

На базе предложенных идей разработана архитектура системы распределённого тестирования. Текущая реализация фреймворка включает в себя транслятор из MSC в C, библиотеки ядра и драйверов и генератор отчетов.

Проведена апробация разработанных методов и программного комплекса. С применением результатов исследования было организовано тестирование в четырех проектах по созданию программного обеспечения в телекоммуникационной области. В итоге было зафиксировано ускорение этапов создания тестового набора, а также сокращение количества тестов, покрывающих требования без снижения количества найденных дефектов.

СПИСОК ЛИТЕРАТУРЫ

1. Beck K. Test-Driven Development by Example. SPb.: Piter, 2003.
2. Fought D.R. Keyword-Driven Testing. Sticky Minds. Software Quality Engineering [электронный ресурс] / URL: <http://www.stickyminds.com/article/keyword-driven-testing> (дата обращения 05.11.2004).
3. ITU-T Recommendation Z.120: Message sequence chart (MSC). Geneva, Switzerland, Oct. 1996.
4. Тютин Б.В., Веселов А.О., Котляров В.П. Масштабирование выполнения тестового набора при автоматизированном тестировании // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2013. № 3(174). С. 118–122.
5. Damm W., Harel D. LSCs: Breathing life into message sequence charts // Formal Methods in System Design. 2001. No. 19(1).
6. Mugridge R., Cunningham W. Fit for Developing Software. London: Prentice Hall, 2005. 384 p.
7. Robot framework [электронный ресурс] / URL: <http://robotframework.org> (дата обращения 25.05.2014).
8. JUnit framework [электронный ресурс] / URL: <http://junit.org> (дата обращения 24.05.2014).
9. UniTesK technology [электронный ресурс] / URL: <http://www.unitesk.ru> (дата обращения 24.05.2014).
10. Тютин Б.В., Никифоров И.В., Котляров В.П. Построение системы автоматизации статической и динамической проверки требований к программному продукту // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб.: Изд-во СПбГПУ, 2012. № 4(152). С. 119–123.
11. HP Quicktest Professional [электронный ресурс] / URL: <http://www8.hp.com/ru/ru/software-solutions/software.html?compURI=1172957> (дата обращения 18.10.2012).
12. IBM Rational Functional Tester [электронный ресурс] / URL: <http://www-01.ibm.com/software/awdtools/tester/functional> (дата обращения 25.05.2014).
13. Hu W., Sun X. Test Case Generation Based on MSC TTCN-3 // Proc. of the Internat. Conf. on Information Engineering and Applications. London: Springer-Verlag, 2013. 888 p.
14. Kaner C., Bach J., Pettichord B. Lessons Learned in Software Testing: A Context-Driven Approach. NY: John Wiley & Sons, Inc., 2001. 320 p.

REFERENCES

1. **Beck K.** *Test-Driven Development by Example*. St. Petersburg: Piter Publ., 2003.
2. **Faught D.R.** *Keyword-Driven Testing. Sticky Minds. Software Quality Engineering*. Available: <http://www.stickyminds.com/article/keyword-driven-testing> (Accessed 05.11.2004).
3. *ITU-T Recommendation Z.120: Message sequence chart (MSC)*. Geneva, Switzerland, October 1996.
4. **Tyutin B.V., Veselov A.O., Kotlyarov V.P.** *Masshtabirovaniye vypolneniya testovogo nabora pri avtomatizirovannom testirovanii, Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye*. St. Petersburg: SPbGPU Publ., 2013, No. 3(174), Pp. 118–122. (rus)
5. **Damm W., Harel D.** *LSCs: Breathing life into message sequence charts. Formal Methods in System Design*, 2001, No. 19(1).
6. **Mugridge R., Cunningham W.** *Fit for Developing Software*. London: Prentice Hall, 2005, 384 p.
7. *Robot framework*. Available: <http://robotframework.org> (Accessed 25.05.2014).
8. *JUnit framework*. Available: <http://junit.org> (Accessed 24.05.2014).
9. *UniTesK technology*. Available: <http://www.unitesk.ru> (Accessed 24.05.2014).
10. **Tyutin B.V., Nikiforov I.V., Kotlyarov V.P.** *Postroyeniye sistemy avtomatizatsii staticheskoy i dinamicheskoy proverki trebovaniy k programmnomu produktu, Nauchno-tekhnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye*. St. Petersburg: SPbGPU Publ., 2012, No. 4(152), Pp. 119–123. (rus)
11. *HP Quicktest Professional*. Available: <http://www8.hp.com/ru/ru/software-solutions/software.html?compURI=1172957> (Accessed 18.10.2012).
12. *IBM Rational Functional Tester*. Available: <http://www-01.ibm.com/software/awdtools/tester/functional> (Accessed 25.05.2014).
13. **Hu W., Sun X.** *Test Case Generation Based on MSC TTCN-3, Proceedings of the International Conference on Information Engineering and Applications*. London: Springer-Verlag, 2013, 888 p.
14. **Kaner C., Bach J., Pettichord B.** *Lessons Learned in Software Testing: A Context-Driven Approach*. NY: John Wiley & Sons, Inc., 2001, 320 p.

ТЮТИН Борис Викторович — аспирант кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: b.tyutin@gmail.com

TYUTIN, Boris V. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.
E-mail: b.tyutin@gmail.com

ВЕСЕЛОВ Алексей Олегович — научный сотрудник кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: veselov.alexey@gmail.com

VESELOV, Aleksey O. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.
E-mail: veselov.alexey@gmail.com

КОТЛЯРОВ Всеволод Павлович — профессор кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: vpk@ics2.ecd.spbstu.ru

KOTLYAROV, Vsevolod P. *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.
E-mail: vpk@ics2.ecd.spbstu.ru