

УДК 004.681

СИСТЕМА ТЕСТИРОВАНИЯ АВИАЦИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА БАЗЕ МЕТОДОВ РАСПОЗНАВАНИЯ ОБРАЗОВ

Козырев В. П., Троицкий А. К.

ФГАОУ ВПО «Национальный исследовательский ядерный университет «МИФИ» (НИЯУ МИФИ), Москва, Россия (115409, Москва, Каширское ш., 31), e-mail: VPKozyrev@mephi.ru, Troitsky@cyber.mephi.ru

Тестирование программного обеспечения (ПО) авиационных систем, формирующих графические изображения, отображающие разнообразную полетную информацию на экране монитора, является весьма трудоемким, но необходимым процессом. Повысить эффективность этого процесса можно за счет автоматизации как планирования испытаний (тестов), так и их выполнения. Основной целью автоматизации тестирования является замена оператора-тестировщика (человека) компьютерной системой анализа изображения, находящегося на экране монитора. Самой сложной задачей, которую приходится решать для достижения этой цели, является создание системы принятия решений по проверке соответствия элементов изображений, выводимых на экран, заданных тест-планами. В работе рассматриваются методы и средства описания изображений, содержащихся на экране монитора, их анализа и принятия решений для их проверки.

Ключевые слова: методы принятия решений, искусственный интеллект, распознавание образов, автоматизация тестирования ПО, графические интерфейсы.

AVIONICS SOFTWARE TESTING SYSTEM BASED ON DECISION MAKING METHODS

Kozyrev V. P., Troitsky A. K.

National research nuclear university MEPHI (NRNU MEPHI), Moscow, Russia (115409, Moscow, Kashirskoye shosse, 31), e-mail: VPKozyrev@mephi.ru, Troitsky@cyber.mephi.ru

One of typical functions of software for airborne systems and equipment is to display various types of flight information on monitor screens in graphical form. Testing of that graphical software is a very time consuming process, but is required for safety of flights. It is possible to improve that process with making it less expensive and more reliable by means of automation of test planning and execution. The main purpose of test execution automation is to replace a human-operator by a computer system, which can recognize and analyze screen images produced by the software under test, and compare them with the expected ones defined in the test plans. The most complex problem, which it is need to be solved for achieving this purpose, is to create decision making system for checking accordance of monitor image elements and test plans. Screen image description, analysis, and decision making methods are considered in the article. A system is described, which can automatically recognize several types of screen images, including texts, scales and pointer indicators.

Key words: decision making methods, artificial intelligence, software testing automation, pattern recognition, graphical interface.

Введение

Важное место при разработке программного обеспечения (ПО) авиационных систем занимает процесс его верификации и, в частности, тестирования. Причем к тестированию авиационного ПО, поскольку от его работы могут зависеть жизни людей, предъявляются очень высокие требования [4]. Для тестирования функциональных систем ПО могут применяться разные методы, требующие различных затрат на их реализацию. Одним из наиболее трудоемких является процесс тестирования ПО, выполняющего вывод на экраны мониторов графической информации.

При тестировании авиационного ПО, выполняющего вывод графической информации, решается задача оценки соответствия выводимых изображений требованиям к ПО, на основе

которых оно реализовано. Одновременно с ней должна решаться и задача установки необходимых значений входных данных, по которым формируются графические изображения. Эти значения вместе с ожидаемыми на выходе системы изображениями определяются при подготовке плана испытаний (тест-плана), содержащего сценарий, по которому эти испытания должны проводиться. Подготовка тест-плана – это еще одна задача, решаемая в связи с двумя первыми.

Одним из возможных подходов к тестированию является решение этих задач «вручную». В этом случае при составлении тест-планов в них необходимо определить как последовательность действий оператора по вводу данных, приводящих ПО в определенные состояния, так и ожидаемые результаты работы ПО. Эти результаты могут задаваться, например, в виде словесных описаний или в виде изображений, которые должны появиться на экране монитора. В исполнение этих тест-планов производится запуск ПО в соответствии с определенными в них сценариями: задаются входные данные, и производится визуальное сравнение результатов, получаемых в ходе работы ПО, с ожидаемыми результатами. Выполнение сценария в этом случае может потребовать постоянного внимания оператора в течение длительного времени. Многочасовые затраты времени для одного сценария испытаний при таком подходе являются обычным явлением. А случайная ошибка оператора, например, при вводе данных, вполне вероятно потребует повтора всего многочасового сценария, если в тестовой среде не реализована возможность «отката» назад по сценарию для исправления сделанной ошибки.

Снизить затраты и повысить надежность процесса тестирования возможно за счет его автоматизации, в частности, обеспечив автоматизацию решения перечисленных выше задач. А поскольку эти задачи являются связанными, больший эффект может быть получен при комплексной автоматизации их решения. Основной целью автоматизации тестирования является замена оператора-человека компьютерной системой анализа изображения, находящегося на экране монитора. Самой сложной задачей, которую приходится решать для достижения этой цели, является создание системы распознавания элементов изображений, выводимых на экран. Такие элементы могут содержать символьные и графические данные. Причем для отображения текстовой информации могут использоваться разные типы и размеры шрифтов, а графические элементы могут иметь произвольный вид, что требует настройки на них в каждом конкретном случае. В связи с этим следует заметить, что адаптивность системы распознавания изображений является необходимым свойством рассматриваемой среды автоматизации тестирования.

Другим общим требованием к разрабатываемой системе является обеспечение ее универсальности, понимаемое в том смысле, что система должна обеспечить организацию

среды тестирования, в которой могут выполняться (тестироваться) различные программы, в том числе и не имеющие каких-либо специальных интерфейсов для взаимодействия с ней.

В основе предлагаемого подхода к автоматизации тестирования графического ПО лежат автоматическое распознавание изображений, выводимых на экран в результате работы тестируемого ПО, и оценка соответствия этих изображений ожидаемым результатам, задаваемым в тест-плане. Общая схема этого подхода, реализованного в разрабатываемой системе автоматизации тестирования, изображена на рис. 1.

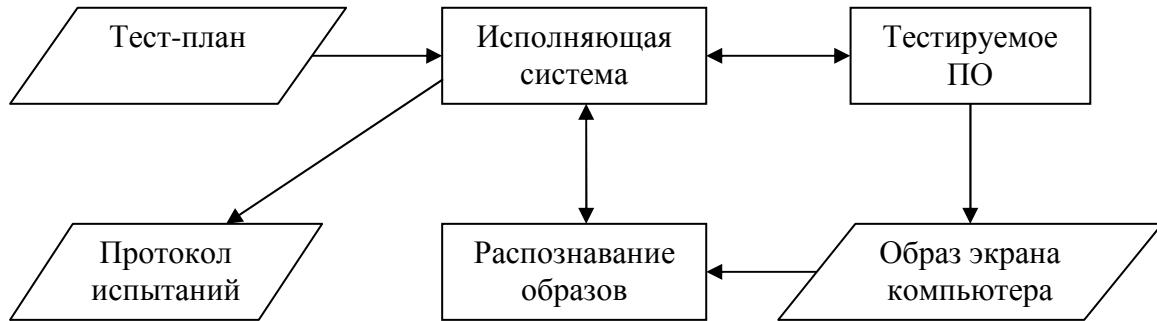


Рис. 1. Схема автоматизированного процесса тестирования

В поступающих на вход системы тест-планах содержатся сценарии, включающие множество шагов тестирования. На каждом шаге в сценарии определяются входные данные для тестируемого ПО и ожидаемые результаты его работы с заданными входами. Базис, на котором обеспечивается автоматизация выполнения сценариев тестирования и интеграция тестируемого ПО в среду тестирования, реализуется в блоке «Исполняющая система». Он выполняет интерпретацию сценариев, обеспечивает передачу заданных входов в тестируемое ПО и сравнивает получаемые от него выходы с ожидаемыми значениями. При этом числовые данные сравниваются непосредственно исполняющей системой, а графические – в блоке «Распознавание образов», которому, с одной стороны, (от исполняющей системы) передаются описания ожидаемых графических образов, а с другой, – образы экрана (или отдельных его областей), формируемые тестируемым ПО. Результаты сравнений отображаются исполняющей системой в протоколах испытаний. Т.е. собственно автоматизация решения указанных выше задач реализуется в блоках «Исполняющая система» и «Распознавание образов», представленных на схеме.

Присутствующий в системе интерфейс взаимодействия с тестируемым ПО позволяет разрабатывать драйверы, обеспечивающие обмен данными с тестируемым ПО и прием образов экрана (в формате bitmap) по различным каналам. В качестве стандартных функций в систему включены возможности моделирования ввода с клавиатуры и мыши для управления тестируемым ПО, а также считывания данных с экрана инструментального

компьютера. Эти функции позволяют тестировать графические приложения, выполняемые под управлением инструментальной операционной системы. В настоящее время реализуется версия системы, работающая под Windows.

Идея использования для тестирования ПО инструментов автоматизации запуска тестируемых программ, обеспечивающих моделирование нажатий клавиш и движения мыши, не является новой. Однако известные инструменты универсального применения, например, AutoIt [3] или Visual Test [5], не обеспечивают возможности определения в сценариях графических образов и, соответственно, возможности автоматизации их анализа. Для обеспечения этой возможности в язык описания сценариев, интерпретируемый в разрабатываемой системе, включены средства описания графических данных, которые позволяют задавать координаты интересующих тестирующего полей (областей экрана), их типы и ожидаемые значения. В настоящее время реализована обработка полей типов «Текст», «Линейная шкала», «Угловая шкала» и «Bitmap». Ожидаемые значения текстовых полей и шкал задаются в текстовом или числовом виде соответственно. При этом в сценарии определяются необходимые атрибуты этих полей (например, цена деления – для шкалы или шрифт – для текста), которые учитываются при обработке получаемых изображений. Для полей типа «Bitmap» в сценариях задаются ссылки на файлы в формате bitmap, которые рассматриваются как ожидаемые значения соответствующих полей.

Фактором, влияющим на трудоемкость подготовки сценариев, являются автоматизация разметки изображений и выделения необходимых полей. В разрабатываемой системе пользователю предоставляется возможность выделения мышью интересующих его областей на экране, которым затем присваиваются имена, определяются их типы и необходимые атрибуты, после чего полученные определения полей могут использоваться в формируемых тест-планах. Следует заметить, однако, что задача автоматизации планирования испытаний (в частности, определения того, что и в какой последовательности проверять) здесь не рассматривается.

Анализ элементов экранного изображения

В качестве объекта автоматизации было выбрано одно из используемых на мониторах большинства самолетов изображений. Оно выводится симулятором на экран монитора инструментального компьютера в виде, аналогичном тому, в котором оно выводится на экран монитора PFD (Primary Flight Display) в кабине самолета.

Для тестирования системы необходимо задать начальные данные – указать на изображениях на экране монитора те компоненты, которые будут контролироваться в процессе работы. С этой целью проводится анализ информации, отображаемой на дисплее, и выбираются те данные, которые изменяются в процессе работы.

Изменяющиеся данные на экране монитора PFD можно разделить на три основные группы – символьные, стрелочные и шкалы. Они приведены на рис. 2.

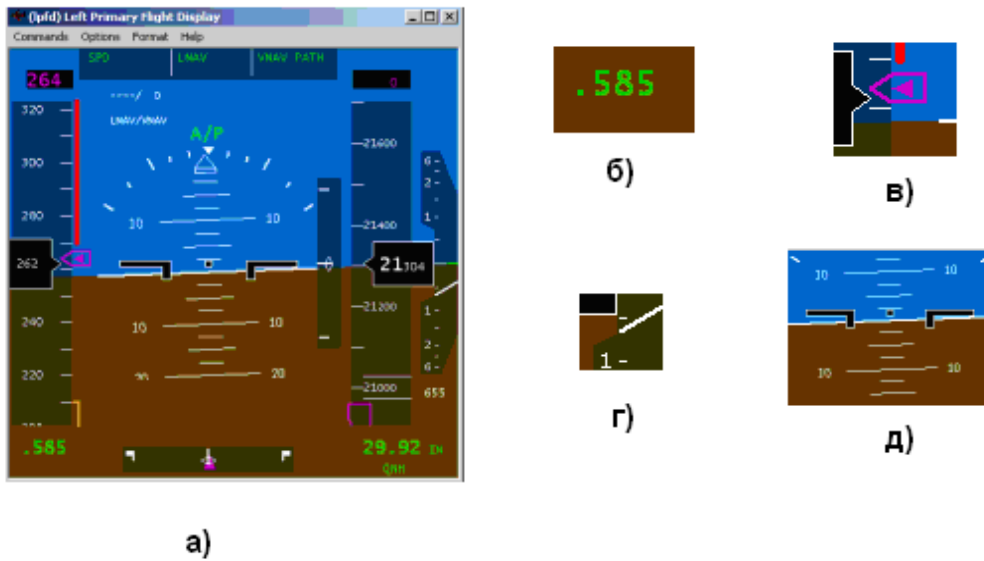


Рис. 2. а) – общий вид PFD; б) – цифровые данные, в) – стрелочный индикатор, г) – боковые шкалы; д) – центральная шкала

При подготовке системы к работе оператор выделяет на экране области, содержащие анализируемые элементы. Для компонент, содержащих символьную информацию, которые не изменяют своего положения на экране, он выделяет границы фиксированных областей. Для стрелочного индикатора, визуально представляющего собой отрезок прямой линии, оператор задает горизонтальные координаты 2-х условных вертикальных прямых линий, которые в каждый момент времени пересекает стрелка индикатора. Анализ центральной и боковых шкал выполняется с помощью задания горизонтальных координат условных вертикальных линий, пересекающих линии разметки шкал. Для шкал также фиксируются размеры областей численных данных, размещенных на шкале, их расположение относительно линий разметки и по которым определяется ее положение, и координаты индикаторного элемента, относительно которого перемещается шкала.

В результате проведенной подготовки к тестированию получается массив данных, содержащих исходную информацию для выполнения тестирования. Для символов это массив данных, количество элементов которого равно числу анализируемых позиций чисел с фиксированным расположением на экране монитора. Каждый элемент массива хранит четыре значения, задающих границы прямоугольника, внутри которого на экране монитора расположено анализируемое числовое значение.

Информация о стрелочных индикаторах задается данными, содержащими координаты двух вертикальных линий, по точкам пересечения с которыми определяется угол наклона

стрелки и горизонтальную координату условной вертикальной линии, на которой заканчивается стрелка индикатора.

Для каждой из двух боковых шкал задаются горизонтальные координаты условной вертикальной линии, пересекающей метки шкалы, смещение относительно этой линии областей расположения числовых значений шкалы и их размеры. Для центральной шкалы задаются координаты двух условных вертикальных линий, пересекающих на краях нулевую линию шкалы, и позиция фиксированного указателя шкалы уровня.

Такие процедуры, являющиеся индивидуальными для каждого типа изображения, выводимого на экран монитора и соответствующего реальным показаниям на борту самолета, выполняются оператором-планировщиком перед началом работы.

Проверка символьной и графической информации

Проверка символьной информации. При анализе символьной информации необходимо учитывать тот факт, что шрифты, применяемые на экране, могут иметь минимальные размеры, различные типы и масштабы и могут иметь наклон при повороте шкал, на которых они размещены. Алгоритм распознавания символьной информации, лежащий в основе проверки анализируемых данных на соответствие данным, представленным в тест-планах, состоит из трех основных частей – бинаризации выделенных областей, выделения отдельных символов и их распознавание.

На этапе бинаризации все пиксели фона, на котором располагается символьная информация, имеющие одинаковое цветное значения, кодируются символом «0». Остальные пиксели кодируются значением «1». Из последовательности цифр с использованием алгоритма формирования односвязных областей производится выделение отдельных символов, из которых состоит число.

Основным и наиболее сложным является этап распознавания. Для каждого из четырех вариантов изображений, приведенных на рис. 2, разработаны специальные алгоритмы.

Распознавание символьной информации. На основе алгоритма «жука» [1] для каждого опознаваемого символа, представляющего собой односвязную область, прослеживается контур и «заливается изнутри» [2]. Далее определяется число Эйлера (число «дыр», находящихся внутри выделенного контура, ограничивающего символ), по которому символы относятся к различным группам. Определение числа Эйлера позволяет распознать цифру 8 и разделить оставшиеся цифры на три группы. Первая группа состоит из цифр 6, 9 и 0, вторая – из цифр 1, 2, 3, 5, 7 и третья содержит одну цифру 4 (из-за различных вариантов ее написания). Цифра 0 выделяется из-за отсутствия выступающих у нее тонких линейных участков. Цифры 6, 9 и 4 (в случае наличия в ней внутренней замкнутой области) различаются на основе использования метрических характеристик. В результате

последовательного выполнения перечисленных процедур распознаются цифры 8, 0, 6, 9 и 4, для которой число Эйлера не равно 0.

Для случая, когда число Эйлера равно нулю, производится анализ символа на соответствии цифрам 1, 2, 3, 5, 7 и 4 (в случае отсутствия в верхней части символа замкнутой области). Классификация выполняется с помощью сканирования символа по вертикали. Анализируется количество пересечений символов вертикальными линиями и число односвязных областей, являющихся частями символа и находящимися по обе стороны от вертикальной сканирующей линии.

Подобный подход является инвариантным к типам и размерам используемых шрифтов и допускает наклон символов в заданных пределах.

Проверка показаний индикаторов и шкал. Для стрелочного индикатора (рис. 2 в) сначала определяется угол наклона по точкам его пересечения с условными вертикальными линиями в заданных горизонтальных координатах. Затем по известным координатам оси вращения, вертикальной линии шкалы и углу наклона определяется искомое значение.

Числовые значения движущихся в вертикальном направлении боковых шкал (рис. 2 г) определяются следующим образом. Известны вертикальные координаты указателя для каждой шкалы. Позиция шкалы определяется с помощью нахождения белых делений шкалы относительно указателя и распознавания численных значений, которым соответствуют эти линии. С этой целью производится поиск белых линий шкалы путем вертикального сканирования изображения в заданной горизонтальной координате. После нахождения линии шкалы производится проверка, есть ли рядом с ней число, указывающее ее значение. В том случае, если число не обнаружено, производится поиск на соседних делениях шкалы (выше и ниже). Затем выделяется область, в которой найдено число. Его значение определяется по аналогии с распознаванием численных данных. Масштаб изображения известен. Проводится пересчет положения указателя относительно опознанного значения белой отметки деления шкалы по известному масштабу изображения.

Угол наклона горизонтальных делений центральной шкалы (рис. 2 д) определяется таким же образом, как и для стрелочного индикатора: вычисляются координаты точек пересечения нулевого деления шкалы с двумя условными вертикальными линиями в определенных позициях и по ним вычисляется угол наклона. Высота определяется как расстояние, измеренное по вертикали от центра черного квадрата, координаты которого известны (нулевого значения) до пересечения с линией крена.

Технология работы системы в процессе тестирования

Процесс тестирования происходит следующим образом. Перед началом работы в тест-планы заносится информация с координатами анализируемых областей с численной

информацией и сканирующих линий. По командам управляющей программы, имитирующей работу оператора-тестировщика, на экране монитора последовательно происходит изменение численных и графических данных. После каждого такого изменения производится захват экранного изображения. Зафиксированный кадр обрабатывается с помощью приведенных выше методов обработки и распознавания изображений. Полученные результаты сравниваются с заданными в тест-планах. В тех случаях, когда данные не совпадают, система фиксирует время (номер кадра) и параметр, по которому результаты не совпали. После окончания работы сформированный системой отчет предоставляется оператору-тестировщику для анализа и принятия решения о дальнейших действиях.

Заключение

Разработана система тестирования графического ПО на базе методов распознавания образов. В зависимости от сложности выполняемых сценариев испытаний эффект от автоматизации процесса их выполнения может составить сотни и тысячи человеко-часов. Предполагается, что разработанная система тестирования окажется востребованной и ее внедрение принесет существенный эффект. При этом необходимо учитывать, что трудоемкость тестирования, несмотря на средства автоматизации, велика, поэтому систему имеет смысл применять только там, где это действительно необходимо, например, в авиации.

Список литературы

1. Дуда Р., Харт П. Распознавание образов и анализ сцен, пер. с англ. – М.: Мир, 1976. – 511 с.
2. Роджерс Д., Адамс Дж. Математические основы машинной графики. – М.: Мир, 2001. – 604 с.
3. AutoIt. Automation and scripting language. URL: <http://www.autoitscript.com/site/autoit> (дата обращения: 18.01.2012).
4. RTCA/DO-178B "Software Considerations in Airborne Systems and Equipment Certification". – 112 p.
5. Thomas R. Arnold, Thomas R. Arnold II. Visual Test 6 Bible. John Wiley & Sons, 1998. – 712 p.

Рецензенты:

Загребаев Андрей Маркоянович, доктор физико-математических наук, профессор, заведующий кафедрой № 22, Научно-исследовательский ядерный университет «МИФИ», г. Москва.

Рыбина Галина Валентиновна, доктор технических наук, профессор, профессор, Научно-исследовательский ядерный университет «МИФИ», г. Москва.