Orientation in Objects

Testen von Eclipse RCP-Anwendungen

) Schulung)

) Beratung)

) Entwicklung)

) Artikel)

AUTOR

Steffen Schäfer
Orientation in Objects GmbH

Veröffentlicht am: 2.4.2007

TESTEN VON ECLIPSE RCP-ANWENDUNGEN

Mit Version 3.0 wurde Eclipse zu einer mächtigen Platform ausgebaut, die als Basis für Eigenentwicklungen genutzt werden kann. Die Rich Client Platform (RCP) bietet sich als Grundlage an, um Plugin basierte Anwendungen zu erstellen, die auf SWT und JFace aufbauen.

Der Einsatz von neuen Technologien wird allerdings oft von der Testbarkeit abhängig gemacht. Tests sind in der modernen Softwareentwicklung ein unverzichtbares Mittel, um die Qualität einer Anwendung sicher zu stellen. Auch grafische Oberflächen sind testbar.

Für ältere Technologien (z.B. Swing) steht schon seit Längerem eine Reihe von Test-Werkzeugen zur Verfügung, die mit unterschiedlichem Funktionsumfang und Preis aufwarten. Allen gemeinsam ist, dass mit ihnen Testabläufe aufgezeichnet und wieder abgespielt werden können, um das gleiche Verhalten auch in späteren Versionen zu garantieren. Doch wie sieht es mit der Testbarkeit von RCP-Applikationen aus, gibt es auch hier entsprechende Programme?

Im Artikel werden drei Werkzeuge vorgestellt, die das Testen von RCP-Applikationen ermöglichen. Es werden die Schritte beschrieben, die notwendig sind, um Tests aufzeichnen und abspielen zu können. Auch die Unterschiede der einzelnen Programme sollen hierbei betrachtet werden.

VORGEHEN

Für jedes der Programme werden im folgenden die Schritte beschrieben, die nötig sind, um eine Testaufzeichnung durchführen zu können. Anschließend sind einige interessante Funktionen der Programme beschrieben, die ihren Funktionsumfang verdeutlichen sollen. Eine vollständige Liste der Features findet sich in der jeweiligen Produktdokumentation.

Das abschließende Kapitel soll sie Unterschiede und Gemeinsamkeiten der Programme verdeutlichen.

DAS TESTSZENARIO

Als Opfer für die Tests mit den einzelnen Programmen kommt Eclipse selbst zum Einsatz. Eclipse ist eine RCP-Anwendung, auf die jeder Zugriff hat, sodass jeder die hier beschriebenen Schritte selbst nachvollziehen kann. Bei allen Tests wird Eclipse 3.2.1 und Java SE 1.5.0 benutzt.

Der Test umfasst das Anlegen und Löschen einer Klasse innerhalb eines Java-Projektes. Dies ist ein recht einfacher Test, der jedoch verschiedene Komponenten und Dialoge umfasst. Beim Anlegen der Klasse müssen einige einfache Textfelder ausgefüllt werden. Ob die Programme auch mit komplexeren Komponenten arbeiten können, zeigt sich dann beim Löschen, welches über das Kontextmenü in einem Baum durchgeführt wird.

AUTOMATED GUI RECORDER (AGR)

AGR wird im Rahmen der Eclipse Test & Performance Tools Platform (TPTP) entwickelt. Das Projekt hat den Status einer "Technology Preview", macht jedoch einen stabilen Eindruck.

INBETRIEBNAHME / TESTAUFZEICHUNG

Zum Betrieb von AGR sind einige weitere Plugins notwendig. Dies sind die "TPTP Platform Runtime", die "TPTP Testing Tools Runtime" und das "Eclipse Modeling Framework (EMF and XSD) SDK". Alle aufgeführten Plugins können von der Downloadseite des TPTP Projektes bezogen werden. Weiter unten auf dieser Seite steht natürlich auch AGR (unter "Automated GUI Recording") zum Download bereit. Zur Installation werden nun alle Plugins in das Eclipse Verzeichnis entpackt und dieses anschließend gestartet.

Als Voraussetzung zum Aufzeichnen von Tests wird nun ein Plugin-Projekt und darin eine Testsuite erstellt. Hierzu wird im "Neu"-Dialog von Eclipse der Dateityp "Test > Test From Recording" angeboten.

Abbildung 1: AGR Testsuite

Ist einmal eine solche Testsuite erzeugt, können einfach auf der Seite "Test Cases" über den Button am oberen Rand Testfälle aufgenommen werden. Für jeden neuen Testfall kann vor der Aufzeichnung im Dialog ein Titel, sowie eine Beschreibung gewählt werden. Zusätzlich muss noch die Perspektive festgelegt werden. Zu Beginn des Tests wechselt Eclipse in die gewählte Perspektive und ein kleines Kontrollfenster erscheint, über das der Test gesteuert wird.

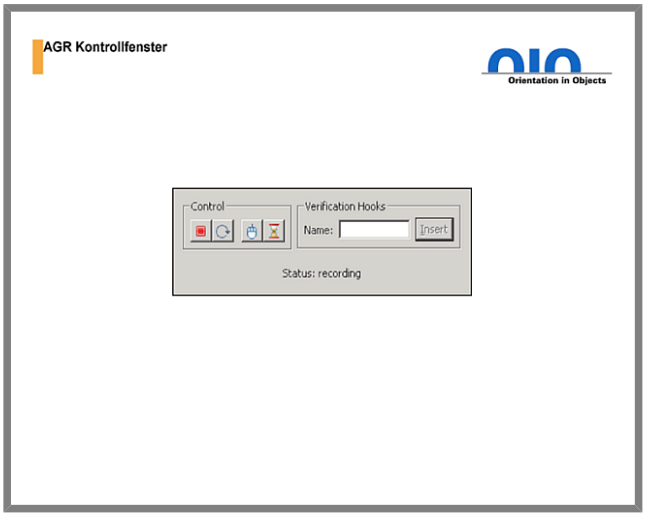

Abbildung 2: AGR Kontrollfenster

Alle nun durchgeführten Aktionen zeichnet AGR für die spätere Wiedergabe auf.

Die einzelnen Testfälle werden innerhalb der Testsuite als XML gespeichert und können mit etwas Übung gelesen und editiert werden. Das Abspielen von Tests wird im einfachsten Fall über einen Knopf am oberen Rand erreicht.

Das oben beschriebene Testszenario konnte aufgezeichnet und ohne Änderungen wieder abgespielt werden.

FEATURES

AGR unterstützt keine automatische Prüfung von Feldinhalten. Alternativ können aber zu jedem beliebigen Zeitpunkt während der Testaufzeichnung sog. Verification Hooks erzeugt werden. Dies sind einfache Methoden innerhalb einer Java Klasse, auf die im XML Code des Tests Bezug genommen wird. Beim Erstellen eines Verification Hooks muss man eine Komponente anklicken. Diese wird beim Aufruf der Testmethode übergeben und dient als Ausgangspunkt für die Validierung.

Beim "Position Based Recording" werden die Aktionen nicht auf die Komponente bezogen aufgezeichnet, sondern Pixelgenau. Hierbei sind nicht die Koordinaten innerhalb des Fensters ausschlaggebend, sondern die Bildschirmkoordinaten. Dies macht jedoch einige Probleme, wenn das Programmfenster nicht immer an der selben Stelle gestartet wird, oder der Test auf einem anderen Betriebssystem (oder Betriebssystemversion) ablaufen soll.

Eine weitere Möglichkeit, die Testaufzeichnung zu beeinflussen, bietet das "Wait Time Recording". In diesem Modus werden alle Wartezeiten des Benutzers mit aufgezeichnet und beim Abspielen des Tests künstlich erzeugt.

BESONDERHEITEN

Beim Abspielen von Tests kann der Benutzer wie gewohnt weiterarbeiten. Alle aufgezeichneten Ereignisse werden beim Abspielen in den Komponenten direkt ausgelöst. D.h. es können keine Seiteneffekte entstehen, wenn das Fenster im Hintergrund oder sogar minimiert ist.

Durch die Integration in TPTP sind beim Testen mit AGR viele Möglichkeiten automatisch gegeben. Es können z.B. Tests aus Ant heraus gestartet werden. Ein Beispiel, wie dies relaisiert wird ist im TPTP Tester Guide [1] zu finden. Auch das Ausführen von Tests auf anderen Rechnern ist mit Hilfe eines Test Execution Service möglich.

LIZENZ

AGR steht unter der EPL (Eclipse Public License) und gilt somit als freie Software. Die Nutzung der Software ist kostenlos. Der Quelltext kann aus dem CVS-Repository des TPTP Projektes bezogen werden.

WINDOWTESTER

WindowTester ist Teil des RCP Developers von Instatiations, einem im Eclipse/RCP Umfeld sehr aktiven Unternehmen. Inzwischen ist WindowTester aber auch als eigenständiges Produkt erhältlich.

INBETRIEBNAHME/TESTAUFZEICHUNG

WindowTester wird als Installationsprogramm ausgeliefert. Während der Installation können die benötigten Plugins in Eclipse eingebunden werden. Dies geschieht automatisch und funktioniert reibungslos.

Vor dem Testen muss ein Plugin-Projekt angelegt werden, um dort die aufgenommenen Tests zu speichern. Die Testaufnahme wird aus der Entwicklungsumgebung über einen zusätzlichen Start-Knopf begonnen.

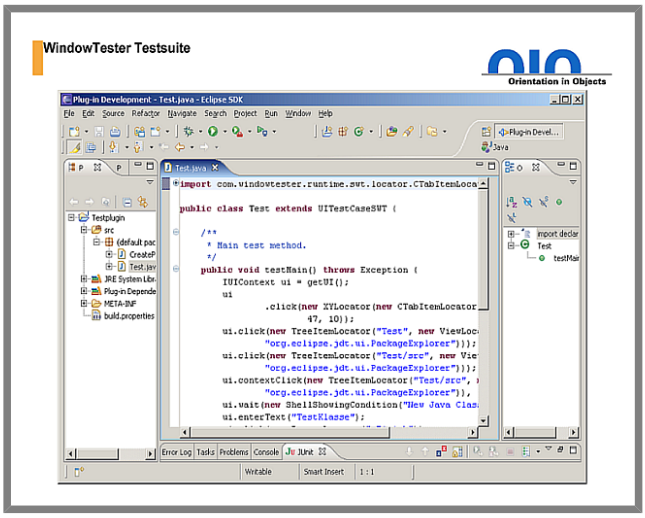

Abbildung 3: WindowTester Testsuite

Mit der startenden Anwendung öffnet sich auch ein Kontrollfenster.

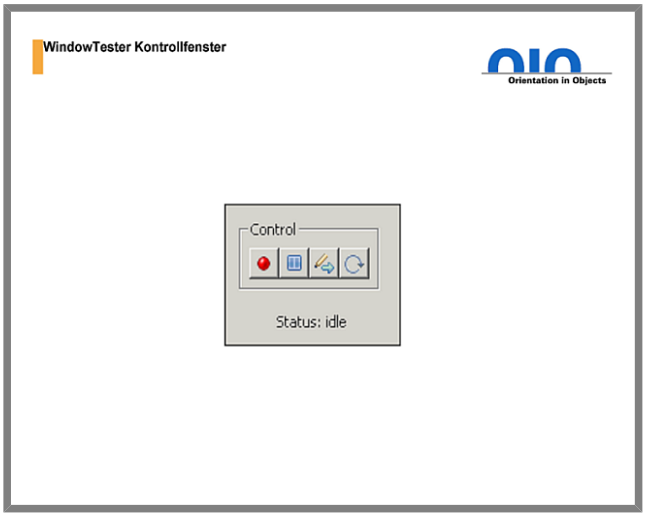

Abbildung 4: WindowTester Kontrollfenster

Die Testaufzeichnung beginnt allerdings nicht direkt, sondern erst nach Auswahl des entsprechenden Knopfes im Steuerfenster. Die nun folgenden Aktionen des Benutzers werden aufgezeichnet.

Nach dem Beenden der Applikation öffnet sich in Eclipse automatisch ein Dialog. Es muss der Name und Speicherort einer Klasse festgelegt werden. In dieser Klasse wird der aufgezeichnete Testablauf in der Methode testMain hinterlegt. Die aufgezeichneten Aktionen sind hier als Aufrufe von Methoden an einen sogenannten UIContext abgelegt. Durch Anpassen der generierten Klasse lässt sich der Testablauf ohne Neuaufzeichnung verändern.

Abspielen lässt sich ein Test durch das Ausführen der Klasse als "Junit Plug-in Test" im Menü "Run > Run as".

Der beschriebene Testfall stellt für den WindowTester kein Problem dar und kann auch hier unverändert abgespielt werden.

FEATURES

Auch WindowTester unterstützt das Erzeugen von Verification Hooks, die hier Assertion Points genannt werden. Die generierten Methoden sind parameterlos. Die Komponenten, die überprüft werden sollen, können bequem über den UIContext ermittelt werden.

Im Einstellungs-Dialog von Eclipse kann für verschiedene Aktionen ein Delay bestimmt werden. Hierdurch ist es möglich einen Test bis zur fehlerhaften Stelle genau zu verfolgen, um so das Fehlverhalten zu analysieren.

Bei den aufgenommenen Tests handelt es sich um JUnit Plugin Tests. Diese können von Ant aus ausgeführt und somit in den meisten Fällen in eine bestehende Testinfrastruktur integriert werden.

BESONDERHEITEN

Beim Ablauf eines Tests muss die Applikation immer im Vordergrund sein, da alle Ereignisse simuliert werden. D.h. der Mauszeiger wird auf dem Bildschirm bewegt und das Drücken eines Knopfes durch einen simulierten Tastendruck ausgeführt. Ist die gesuchte Komponente aber überlagert, so kann sie nicht gefunden werden und der Test bricht erfolglos ab. Dieses Verhalten kann in Kombination mit einem Delay jedoch zum Debuggen verwendet werden.

LIZENZ

WindowTester ist ein kommerzielles Produkt. Eine Lizenz ist auf der Homepage erhältlich. Auch eine kostenlose 14 Tage-Testversion steht hier zum Download bereit.

QF-TEST 2.0

QF-Test ist die Weiterentwicklung von qftestJUI, einem auf dem Gebiet der Swing-Tests schon länger bekannten Produkt. In der neuen Version werden nun auch SWT- bzw. RCP-Anwendungen unterstützt.

INBETRIEBNAHME / TESTAUFZEICHUNG

QF-Test wird als Installationsprogramm ausgeliefert und läuft als eigenständige Software. Um SWT-Applikationen testen zu können, muss die swt.jar ausgetauscht werden. Bei RCP-Anwendungen ist der Austausch des swt-Plugins nötig. Diese Vorgehensweise ist durch die Architektur von QF-Test notwendig und hat laut Hersteller keinen Einfluss auf das Verhalten der RCP-Anwendung.

Beim Start von QF-Test erscheint direkt eine leere Testsuite, die in den folgenden Schritten gefüllt wird. Im Menü Extras wird über den Schnellstart Wizard das zu testende Programm eingebunden. Das zu testende Programm wird bei QF-Test als SUT (Software unter Test) bezeichnet. Eine Eclipse basierte Anwendung kann entweder über die eclipse.exe oder die startup.jar gestartet werden. Beide Möglichkeiten können hier genutzt werden. Durch Abschließen des Wizards wird Im Baum unter Extrasequenzen eine sog. Startsequenz erstellt.

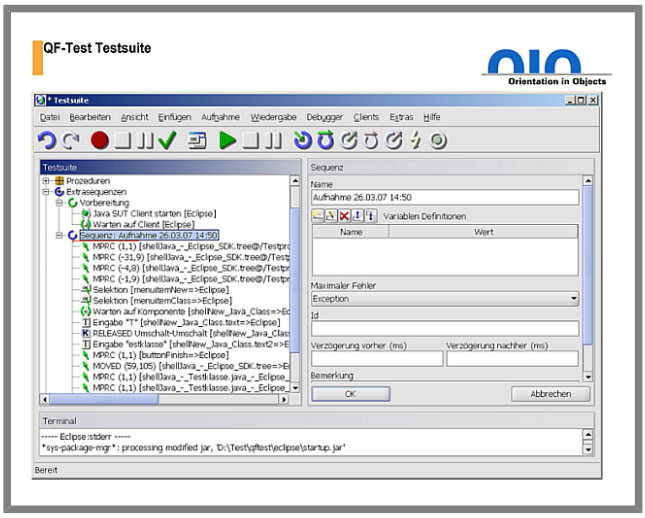

Abbildung 5: WindowTester Testsuite

Beim Ausführen der Startsequenz wird das Programm gestartet. Über einen Knopf in der Toolbar kann nun die Aufzeichnung begonnen werden. Beim Beenden der Testaufzeichnung wird auch der Ablauf des Tests unter Extrasequenzen hinterlegt und kann bei Bedarf bequem editiert werden.

Ohne das Testprogramm zu beenden können die aufgenommenen Sequenzen abgespielt, oder neue aufgenommen werden.

Das Aufzeichnen und Abspielen des Testszenarios stellt für QF-Test kein Problem dar und kann ohne manuelle Änderungen des Testablaufs durchgeführt werden.

FEATURES

QF-Test kann die Inhalte von angeklickten Feldern automatisch überprüfen. Der Button, um dies zu aktivieren, befindet sich im Hauptfenster und trägt den Namen "Checks aufnehmen".

Jedem Schritt eines Tests können künstliche Verzögerungen hinzugefügt werden, um z.B. kritische Stellen in langsamerer Geschwindigkeit verfolgen zu können.

BESONDERHEITEN

Wie bei AGR ist es auch beim Abspielen von Tests mit QF-Test möglich, wie gewohnt weiter zu arbeiten. Alle Events werden in der ausgetauschten SWT-Bibliothek erzeugt. Der Anwender behält also die Kontrolle über die Maus.

QF-Test beherrscht die Modularisierung von Testsuites. Im einfachsten Fall ist dies die Aufteilung eines Tests in mehrere Sequenzen. Darüber hinaus können auch Prozeduren für häufig verwendete Abläufe oder Prüfungen erstellt werden. Hierbei bietet QF-Test auch Möglichkeiten wie Parameterübergabe, Fehlerbehandlung und Aufrufe in anderen Testsuites.

Beim Testen ist es oft hilfreich, den selben Ablauf mit unterschiedlichen Werten durchlaufen zu lassen. Dies kann in QF-Test mit Datentreibern realisiert werden. Die Datensätze werden hierbei in eine Datentabelle eingegeben, aber auch CVS Dateien können eingebunden werden.

Durch Integration der Skriptsprache Jython können z.B. Prüfungen und Testabläufe realisiert werden, die eine komplexere Logik erfordern.

Durch Ausführen mehrerer Instanzen eines Tests auf verschiedenen Rechnern können auch Lasttests durchgeführt werden.

Auch die Dokumentation spielt bei QF-Test eine große Rolle. Die einzelnen Knoten haben ein Feld Bemerkung. Insbesondere bei Prozeduren und Testfällen, etc macht es Sinn, dieses zu nutzen, da sich hieraus Dokumentationen im HTML Format genieren lassen. Die Ergebnisse von Testläufen können ebenfalls als HTML oder XML gespeichert werden, was speziell im Batchbetrieb sehr hilfreich ist.

LIZENZ

QF-Test ist in verschiedenen Ausführungen auf der Firmenhomepage von qfs erhältlich. Eine zeitlich uneingeschränkte Testversion, mit der allerdings keine Tests gespeichert werden können, steht ebenfalls zum Download bereit. Eine funktional uneingeschränkte Testversion kann über ein Formular auf der Homepage bezogen werden.

FAZIT

Allen Programmen gemeinsam ist die Fähigkeit, Abläufe einer grafischen Oberfläche aufnehmen und abspielen zu können. Dies geht mit allen Kandidaten initial recht schnell und lädt zum Testen ein. Für jedes der Programme gibt es ein Einstiegstutorial, welches im Fall von QF-Test sogar sehr umfangreich ausgefallen ist. Auch mit seiner ausführlichen Dokumentation kann QF-Test glänzen.

Ein weiterer Punkt, der betrachtet werden muss, ist der Support. Im Fall der beiden kommerziellen Produkte QF-Test und WindowTester helfen die Hersteller bei Fragen gerne und unterstützen den Anwender bei Problemen. Bei AGR ist die Mailingliste des TPTP Projektes der Hauptanlaufpunkt, um Probleme zu klären.

AGR ist unter der EPL lizenziert und kann somit an die eigenen Bedürfnisse angepasst und erweitert werden. Will man dies nicht tun, so kann dieser Punkt allerdings auch nicht als Vorteil gewertet werden.

Es besteht ein großer Unterschied zwischen QF-Test und den anderen beiden Programmen. QF-Test ist ein Komplettpaket, das sich um alle Details des Testens kümmert. Hierzu gehört z.B. die Generierung von Dokumentationen und das durchführen von Tests im Batchbetrieb. Um dies zu leisten, bedienen sich WindowTester und AGR anderer Software, wie Ant, Junit und TPTP. Der Vorteil, den QF-Test hierbei bietet ist die zentrale Dokumentation, in der alle Teile der Software besprochen werden. Andererseits kennen sich viele Entwickler mit Ant und Junit aus und benötigen hier nur selten eine Dokumentation.

Der erhoffte Nutzen dieses Artikels als Appetithappen auf den Test von Eclipse RCP Anwendungen dürfte nun erbracht sein. Um sich einen Eindruck der Tools zu verschaffen, bieten sich die kostenlosen Testversionen an. Da sich die von den Werkzeugen angebotenen Funktionen und die benötigten finanziellen Mittel stark unterscheiden, sollte sich nun natürlich jeder Tester ein systematisches Bild für seine Entscheidung machen. Gerne können Sie auch uns für eine weitergehende herstellerneutrale Meinung zu Rate ziehen.

REFERENZEN

- [1] TPTP Tester Guide
  http://help.eclipse.org/help32/index.jsp
  (http://help.eclipse.org/help32/index.jsp?topic=/org.eclipse.hyades.test.doc.user/tasks/texecuting-the-test-execution-service.htm)
- Quality First Software GmbH
  (http://www.qfs.de)
- Homepage des TPTP Projektes
  (http://www.eclipse.org/tptp/)
- Instantiations, Inc.
  (http://www.instantiations.com)
- Einführung in AGR
  (http://www.eclipse.org/tptp/test/documents/userguides/Intro-Auto-GUI-4-3-0.html)
- Dokumentation von WindowTester
  (http://downloads.instantiations.com/WindowTesterDoc/integration/latest/docs/html/toc.html)
- AGR Demo
  (http://www.eclipse.org/tptp/test/documents/userguides/auto-gui-viewlet/auto-gui-viewlet_viewlet_swf.html)