Mapas de navegación para la automatización de pruebas de aceptación en aplicaciones móviles

Ginger Janet Valencia-Vásconez[1], Miguel Á. Latre[1] (0000-0002-6682-8383), F. Javier López-Pellicer[1](0000-0001-6491-7430), Javier Nogueras-Iso[1](0000-0002-1279-0367)

[1] Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza
neyitagin@gmail.com, latre@unizar.es, fjlopez@unizar.es, jnog@unizar.es

Resumen. Para que el proceso de pruebas del *software* resulte eficaz es habitual que tanto la parte proveedora como la parte aceptante colaboren. Esto es especialmente cierto en el nivel de las pruebas de aceptación, donde la responsabilidad principal recae en la parte aceptante. Aunque se ha avanzado mucho en herramientas que facilitan de forma automática la generación de casos de prueba y la ejecución de los mismos, estas están normalmente pensadas para desarrolladores con avanzados conocimientos en programación. Este trabajo presenta un componente, denominado Graph2Test, que pretende facilitar la comunicación entre la parte aceptante y la parte desarrolladora en base a la representación mediante un diagrama de estados de los mapas de navegación de la aplicación objeto de las pruebas. A partir de este mapa de navegación se generan casos de prueba en texto plano (lenguaje Gherkin), que posteriormente se transforman en *scripts* de prueba utilizando la tecnología Cucumber. El componente está especializado en la prueba de aplicaciones móviles Android y autocompleta parcialmente estos *scripts* de prueba con instrucciones del entorno de automatización Espresso.

Palabras clave. Testing, pruebas de aceptación, automatización de pruebas, Android, Cucumber, Espresso

1 Introducción

El proceso de pruebas de *software* es un proceso mediante el cual se aplican una serie de métodos (ocasionalmente utilizando herramientas) que permiten obtener un conjunto de medidas para verificar y validar el funcionamiento requerido del *software*. Las pruebas de *software* pueden ser ejecutadas por cualquiera de las partes implicadas en el desarrollo de *software*, ya sea la parte proveedora (por ejemplo, la empresa que lo desarrolla), la cual se asegura de que el producto entregado cumple con todos los requerimientos necesarios, o la parte aceptante (cliente o usuario para el cual se realiza el desarrollo de *software*), la cual se asegura de que el producto recibido es lo que se había solicitado.

Podemos distinguir distintos niveles de prueba, asociados a las responsabilidades de cada una de las partes. Así, se definen por la parte proveedora las pruebas de desa-

rrollo, que se aseguran del cumplimiento de especificaciones técnicas y las pruebas de sistema, que comprueban el cumplimiento de requisitos y de diseño técnico. Por la parte aceptante tenemos las pruebas de aceptación, las cuales verifican que el producto cumple con las expectativas.

Aunque se ha avanzado en la generación automática de casos de prueba y la ejecución de los mismos, este grado de automatismo es menor en pruebas de aceptación de usuario, ya que las plataformas y herramientas existentes están pensadas normalmente para ser utilizadas por desarrolladores con avanzados conocimientos en programación e ingeniería del *software*. De hecho, ya que uno de los pocos elementos sobre los que se llega a un acuerdo entre cliente y desarrollador durante la fase de análisis de una aplicación son los prototipos la interfaz y la manera en la que se navegará por ella, las herramientas de automatización que intentan implicar a la parte aceptante en el proceso de pruebas se basan en la utilización de diagramas que modelan el comportamiento e interacción con el usuario de la aplicación, véanse por ejemplo herramientas como Testar [1] o NDT-Suite [2].

Este trabajo presenta un prototipo denominado Graph2Test para facilitar la automatización de pruebas de aceptación de aplicaciones móviles a partir de los mapas que definen su navegación. Aparte de que su diseño está pensado específicamente para plataformas Android, la principal contribución de este trabajo es la integración de tecnologías de código abierto y el facilitar la involucración de la parte aceptante en la elaboración de escenarios modificables sobre texto plano mínimamente controlado.

2 Automatización del diseño y ejecución de casos de prueba de aceptación

Como se ha comentado en la introducción, los mapas de navegación son un elemento clave para asegurar que todos los requisitos establecidos con el cliente se han cumplido al final del desarrollo. Para representar dichos mapas de navegación en este prototipo de forma que puedan servir de entrada para el mismo, se ha optado por el uso de diagramas de estado. Esto permite generar casos de prueba utilizando la técnica transición de estados (*State transition testing* [3]), con un nivel de cobertura de transiciones. Esta técnica aplica un algoritmo de recorrido de grafos para generar un conjunto de casos de prueba que cubran todas las transiciones del diagrama, de acuerdo con el nivel de cobertura elegido.

La figura 1 muestra el flujo de trabajo propuesto para la automatización del diseño y ejecución de las pruebas de aceptación utilizando Graph2Test. En el diagrama de flujo están marcados en cursiva y fondo gris los pasos que requieren intervención manual (parte aceptante o parte proveedora) y los pasos con fondo blanco indican la parte automática ejecutada por Graph2Test. Para utilizar Graph2Test es necesario disponer de un diagrama de estados que represente el diagrama de navegación de la aplicación que se desea probar. Se ha optado por crear este diagrama de estados utilizando la aplicación YEd Graph Editor[1], utilizando algunas convenciones en cuanto al

[1] https://www.yworks.com/products/yed

nombrado de nodos y aristas para facilitar la escritura posterior del código de los *scripts* de pruebas. El grafo se recorre cubriendo todas sus transiciones con la herramienta GraphWalker[2]. A partir de este recorrido y tomando como referencia los textos de cada arista y cada nodo del grafo, se genera la definición de los escenarios en texto plano mínimamente controlado conforme con el lenguaje Gherkin (incluye un conjunto reducido de palabras reservadas), los cuales podrían ser revisados y mejorados por la parte aceptante.

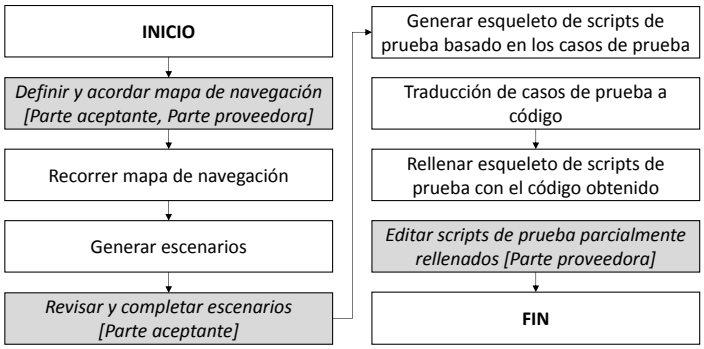

Fig. 1. Diagrama de flujo de trabajo para el diseño y automatización de pruebas.

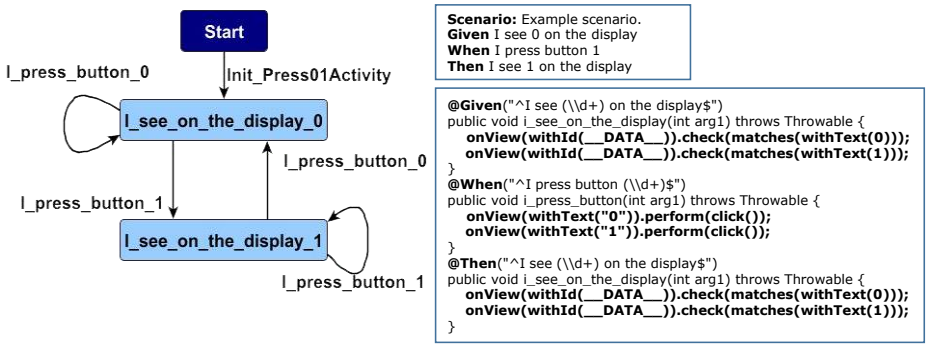

Fig. 2. Ejemplo de modelado de mapa de navegación con YEd Graph Editor, transformación de uno de los escenarios con Gherkin, y rellenado parcial de algunas funciones con Espresso

Una vez obtenidos los escenarios y utilizando la tecnología propuesta por la plataforma Cucumber, se derivan los *scripts* para automatizar las pruebas, generando únicamente un esqueleto (o conjunto de *steps*) de estos *scripts.* Con el objetivo de hacer una implementación parcial de estos esqueletos, se realiza además una traducción de los datos disponibles en el diagrama de navegación a instrucciones de Espresso (o el código Java oportuno). Espresso es una tecnología diseñada especialmente para las pruebas de interfaz de usuario de aplicaciones Android.

La figura 2 muestra a la izquierda el diagrama de estados de una aplicación objeto de pruebas creado con YEd, y a la derecha uno de los cinco escenarios del fichero

[2] http://graphwalker.github.io/

feature obtenido por el prototipo Graph2Test al recorrer el grafo. También se muestran a la derecha algunas funciones del código generado, parcialmente rellenado con código Espresso, en las que queda pendiente editar y completar aquellos detalles que no se han podido automatizar (en este caso, utilizando una instrucción condicional para ejecutar una u otra de las instrucciones Espresso en función del parámetro).

3 Conclusiones y trabajo futuro

Con el componente presentado en este trabajo no solo se ha pretendido la ejecución automática de pruebas para aplicaciones móviles Android gracias a la tecnología Espresso, sino, sobre todo, investigar cómo facilitar y automatizar también el diseño de las mismas a partir de diagramas de estados de la aplicación objeto de pruebas y la generación de escenarios compatibles con la tecnología Cucumber, los cuales pueden ser modificados por la parte aceptante al utilizar un lenguaje en texto plano mínimamente estructurado.

El prototipo de componente se ha probado con dos aplicaciones que han sido utilizadas como objeto de estudio para para comprobar la viabilidad de la infraestructura de pruebas desarrollada. La primera es una pequeña calculadora (aplicación Cukeulator[3]) que realiza funciones de cálculo básicas y muestra los resultados por pantalla. La segunda es la aplicación Farmacias Ahora! ZGZ[4], que muestra en tiempo real información sobre las farmacias existentes en la ciudad de Zaragoza y su estado de apertura, cierre o guardia. Como primera aproximación, se ha conseguido desarrollar un prototipo con funcionalidades básicas, así como la identificación de una serie de problemas y de posibilidades de mejora. Entre las mejoras para abordar como trabajo futuro se consideran las siguientes: simplificación de los diagramas a través de algún mecanismo de parametrización; modificaciones en el diagrama que mejorarían la generación de escenarios de Cucumber; posibilidad de generar escenarios múltiples en Cucumber; conversión de Graph2Test en un plug-in para Android Studio; y utilizar también como entrada un listado de los recursos utilizados en la aplicación y de su ID.

Referencias

1. Vos, T. E., Kruse, P. M., Condori-Fernández, N., Bauersfeld, S., Wegener, J.: Testar: tool support for test automation at the user interface level. International Journal of Information System Modeling and Design (IJISMD), 6(3), 46-83 (2015)
2. García-García, J.A., Ortega, M.A., García-Borgoñon, L. Escalona, M.J.: NDT-Suite: a model-based suite for the application of NDT. In: 12th International Conference on Web Engineering, pp. 469-472 (2012)
3. ISO/IEC/IEEE: ISO/IEC/IEEE 29119-4:2015. International Standard for Software and systems engineering - Software testing - Part 4: Test techniques (2015)

[3] https://github.com/cucumber/cucumber-jvm/tree/master/examples/android/android-studio/Cukeulator

[4] http://www.geoslab.com/es/product/farmacias-ahora-zgz