

Agente Softbot para Pruebas Automáticas de GUI

Lucas Martín Prado⁽¹⁾ Carlos Casanova^(1,3) y Fabiana Piccoli^(2 y 3)

⁽¹⁾ Univ. Tecnológica Nacional, Facultad Regional Concepción del Uruguay, Concepción del Uruguay

⁽²⁾ LIDIC- Univ. Nacional de San Luis, San Luis

⁽³⁾ Universidad Autónoma de Entre Ríos, Concepción del Uruguay
Argentina

{pradol, casanovac}@frcu.utn.edu.ar

mpiccoli@unsl.edu.ar



Contexto

Esta propuesta de trabajo se lleva a cabo dentro del proyecto de investigación: "Analíticas basadas en incertidumbre para soportar la toma de decisiones en la ingeniería de software" (UTN).

Líneas de Investigación

Pruebas basadas en interfaz gráfica (GUI)

Las pruebas basadas en la interfaz gráfica de usuario hacen referencia a las pruebas de sistema realizadas a través de la GUI del sistema. A diferencia de pruebas de bajo nivel de abstracción, donde se puede determinar qué línea de código es la que provoca el error, estas pruebas con mayor nivel de abstracción solo responden qué funcionalidad presenta defectos en el sistema, por lo que requieren un posterior análisis del software bajo prueba (SUT, Software Under Test) para encontrar la causa del defecto.

Clasificación de pruebas basadas en GUI

Existen múltiples técnicas para las pruebas basadas en GUI, a las cuales se las clasifica en distintas generaciones. La primera generación, también conocida como las pruebas basadas en GUI por coordenadas, utilizan las coordenadas exactas en la pantalla para interactuar con la GUI del SUT. Esta técnica se abandonó debido a que las GUI de los programas suelen cambiar con el tiempo, lo que provocaba que las pruebas precisen mucho esfuerzo de mantenimiento. La segunda generación, conocida también como pruebas basadas en GUI por componentes, es una técnica que aún a día de hoy continúa vigente y puede utilizarse para pruebas a nivel de sistema. Es una técnica de caja gris y utiliza propiedades de los componentes de la GUI para identificar e interactuar con ellos, lo cual le brinda mayor grado de robustez respecto de los cambios de la GUI. Sin embargo, como las pruebas utilizan información de los componentes de la GUI, cambios al código fuente del programa pueden causar que fallen ciertas pruebas. Otro problema de esta generación es que existen algunos componentes que no pueden ser probados. La tercera generación, también conocida como pruebas visuales de GUI, es una técnica emergente que utiliza emparejamiento de plantillas para identificar e interactuar con la GUI a través de los mapas de bits de las imágenes mostradas al usuario durante la ejecución. La interacción es realizada utilizando el puntero y el teclado, emulando la interacción de un usuario con el SUT. Esto permite que la técnica se utilice independientemente de la implementación o plataforma en la que se ejecute el sistema. Es una técnica de caja negra y no presenta las limitaciones mencionadas en la segunda generación.

Resultados obtenidos/esperados

Se ha creado un dataset de imágenes con etiquetas para los distintos componentes GUI encontrados en páginas web y aplicaciones móviles. Además, se están entrenando distintos modelos de detección de objetos para reconocer dichos componentes.

A futuro se planea:

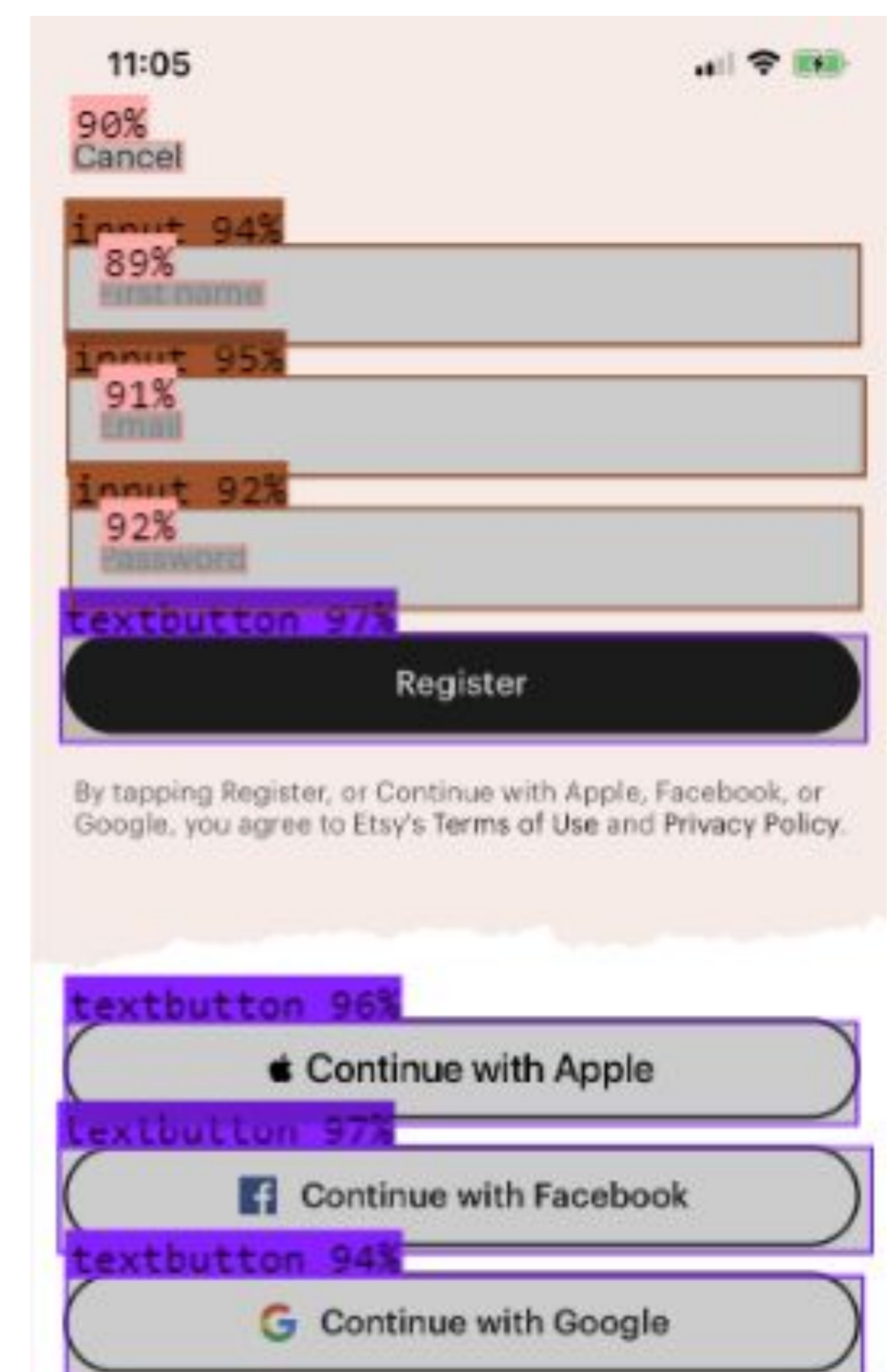
- Formular un modelo conceptual de exploración de GUI y un lenguaje de especificación de acciones.
- Formular diversas estrategias de testing, las cuales constituirán el programa del agente, dependiendo de la tarea de testing requerida y arquitecturas solidarias.
- Implementar el agente en un prototipo de software y realizar pruebas exhaustivas.

Ventajas y desventajas de cada generación

Las técnicas de tercera generación buscan imitar a un usuario humano, por lo que al utilizar como guía las imágenes del monitor fallan las pruebas donde los componentes son invisibles, no existen, o están escondidos (tal como le pasaría a una persona). Si bien esto es un efecto beneficioso, pues al simular la interacción humana son un mejor indicador de la respuesta esperada de un usuario final, estas técnicas tienden a presentar mayor cantidad de falsos positivos para las pruebas de sistema (fallan pruebas que deberían de haber sido superadas). Por el contrario, las técnicas de segunda generación pueden superar las pruebas donde los componentes son invisibles o están escondidos (pruebas que un humano fallaría), pues interactúan directamente con el componente a nivel de código, por lo que reportan un mayor número de falsos negativos en las pruebas de aceptación. Las herramientas que utilizan técnicas de segunda generación tienen problemas para detectar defectos que una persona detectaría fácilmente, como, por ejemplo, la ausencia de un componente en pantalla.

Propuesta

En este trabajo proponemos una línea de investigación que tiene por objetivo último diseñar un agente inteligente del tipo softbot que realice pruebas automáticas de GUI mediante distintas estrategias y que pueda ejecutarse independientemente de la plataforma utilizada por el software bajo prueba. Para ello se propone la utilización de nuevas técnicas en el campo de la Visión Computacional e Inteligencia Artificial, de tal manera que el agente sea independiente del proceso de desarrollo del sistema y pueda, por lo tanto, realizar el proceso de testing sin los sesgos que surgen de la autoría del código a probar.



Formación de recursos humanos

El resultado esperado más importante respecto a la formación de recursos humanos es el desarrollo de una tesis de doctorado. Además, para la confección del dataset de imágenes de componentes GUI se contó con el trabajo de 10 becarios de investigación y se espera que la temática sirva de plataforma para continuar con la formación de becarios alumnos y graduados que quieren iniciarse en la investigación.