

Generación de datos de prueba en el contexto de la Automatización Robótica de Procesos ^{*}

A. Martínez-Rojas, J.M. López-Carnicer, J.G. Enríquez, and A. Jiménez-Ramírez

Departamento de Lenguajes y Sistemas Informáticos. Escuela Técnica Superior de Ingeniería Informática. Avenida Reina Mercedes, s/n. 41012, Sevilla.

{amrojas, jlopez16, jgenriquez, ajramirez}@us.es

Abstract. Un problema común en muchos ámbitos (e.g., automatización robótica de procesos, minería de procesos, o inteligencia artificial) es la obtención de registros de eventos que incluyan capturas de pantalla de la interfaz de usuario (i.e., UI Log). En este artículo se presenta AGOSUIRPA, una plataforma de código abierto para la generación automática de UI Logs y capturas de pantalla tomando como referencia la ejecución de un proceso base. Además, esta herramienta también ofrece la posibilidad de configurar ciertos parámetros permitiendo múltiples opciones de variabilidad en el conjunto de datos generado. AGOSUIRPA supone un gran avance para la comunidad científica e industrial ya que permite reducir los esfuerzos a la hora de realizar experimentación, casos de estudio, o pruebas, sobre soluciones RPA.

Keywords: Robotic Process Automation · Event Log Generation · Task Mining · User Interface Derivation

1 Introducción

En la actualidad existe un interés creciente por la automatización robótica de procesos (RPA, de sus siglas en inglés) así como por la minería de procesos [2, 5, 4]. Aunque existe alguna propuesta que plantea automatizar la ejecución de las pruebas en este contexto [3], sigue siendo necesaria la obtención de datos relativos a la ejecución del proceso para poder estudiar el cumplimiento de los requisitos establecidos o realizar pruebas. Estos datos son difíciles de conseguir, ya que la compartición de datos de procesos de negocio con terceros podría suponer una vulneración de la privacidad de los datos de los usuarios finales (e.g., datos bancarios, direcciones o documentos de identificación). Ante esta situación, han aparecido distintas aproximaciones como [1] para la generación de datasets, o [6] para producir capturas de pantalla para la generación de webs, pero ninguna se aplica al ámbito de procesos para la obtención de UI Logs, es decir, de una serie

^{*} Esta investigación ha sido apoyada por el proyecto NICO (PID2019-105455GB-C31) del Ministerio de Ciencia, Innovación y Universidades de España y el Programa de becas FPU, concedido por el Ministerio de Educación y Formación Profesional de España (FPU20/05984).

de eventos identificados en el tiempo (por ejemplo, clics de ratón y pulsaciones de teclado), obtenidos mediante la monitorización y grabación de las interfaces de usuario.

En este artículo se presenta *AGOSUIRPA*¹, una plataforma para la generación automática de UI Logs y capturas de pantalla, en base a una *semilla* (i.e., un UI log de tamaño mínimo que luego será extrapolado) y una configuración dispuesta por el usuario. Estos datos de prueba pueden ser utilizados por la comunidad científica para la ejecución de experimentos, o bien, para generar un conjunto de pruebas para realizar el testing de robots RPA.

2 AGOSUIRPA

2.1 Plataforma

Para la generación de UI Logs sintéticos, el usuario debe especificar como entrada (1) un log semilla, que incluirá una serie de eventos por cada variante existente en el proceso, (2) el tamaño del log a generar y (3) un archivo de configuración que indique por cada variante y actividad: (i) las funciones de variabilidad (i.e. funciones generan variaciones sobre las capturas o datos del UI Log según sean parámetros de entrada) a emplear, (ii) los parámetros de entrada de dichas funciones y, (iii) en caso de necesitar modificar un componente de interfaz gráfica de usuario (GUI), los componentes GUI a emplear en las variaciones. Con el fin de facilitar la configuración, tanto los parámetros como los componentes GUI han sido catalogados. La configuración se realiza a través de un archivo en formato JSON. Este archivo, se puede generar desde cero siguiendo la plantilla disponible en el repositorio, o a través de un asistente para generarlo de forma interactiva.

La generación de UI Logs consta de dos niveles de abstracción. El más básico es a *nivel de caso* (i.e., la ejecución completa de una variante, por tanto comprende las acciones realizadas por el usuario durante la ejecución de una instancia del proceso.), que consiste en realizar variaciones sintéticas asociadas a la acción de un humano sobre el sistema (e.g., cambiar el contenido de un input de texto o la imagen adjunta de un correo). El más abstracto es a *nivel de escenario* (i.e., el propio ecosistema que va a tener un usuario durante todos los casos que tenga el UI Log, es decir, el *look and feel* de las aplicaciones que use), que se trata de una fase anterior a la generación a nivel de caso, en la que, a partir de la semilla, se generan tantas semillas como escenarios adicionales se especifiquen. AGOSUIRPA permite especificar variaciones de botones, de estilo del texto, de los campos de los formularios, etc.

La plataforma ofrece estos dos niveles de abstracción a través de una entidad denominada experimento, donde se almacenará la siguiente información:

- **Nombre:** texto con el nombre del experimento.
- **Descripción:** texto con la descripción del experimento.
- **UI Log semilla:** archivo CSV base para la configuración de UI Logs sintéticos.
- **Capturas de pantalla:** en caso de generar variabilidad en capturas, archivo comprimido con las capturas del UI Log semilla.

¹ Código: <https://github.com/RPA-US/agosuirpa>

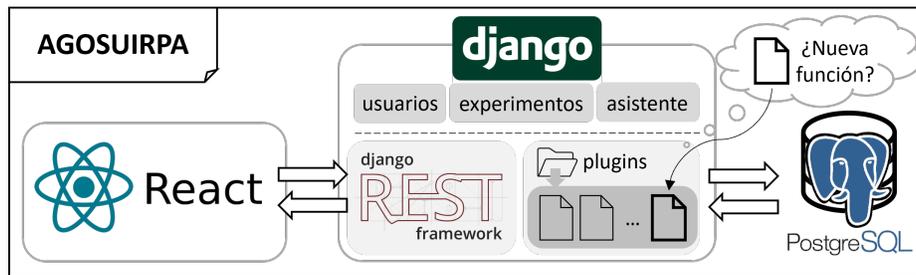


Fig. 1. Arquitectura del sistema

- **Número de escenarios adicionales:** número de escenarios a generar en caso de querer crear escenarios adicionales al base.
- **Variabilidad a nivel de escenario:** archivo JSON de configuración con la variabilidad a nivel de escenario.
- **Tamaño del UI Log:** número de eventos contenidos en el UI Log a generar.
- **Porcentajes para UI Logs imbalanceados:** por cada tamaño se genera un log balanceado (con el mismo número de instancias por variante) y otro imbalanceado con el porcentaje de instancias de cada variante indicado.
- **Variabilidad a nivel de caso:** archivo JSON de configuración con la variabilidad a nivel de caso.

Una vez configurado y ejecutado el experimento, se obtendrán tantos UI Logs sintéticos como tamaños y número de escenarios se hayan indicado. Cada variación configurada por el usuario se almacena en base de datos para mantener la trazabilidad de los cambios y para su posterior consulta. Cuando finaliza el proceso de generación, quedará disponible para su descarga un archivo comprimido con todos datos de prueba generados, i.e., los UI Log generados en formato CSV junto con las capturas de pantalla asociadas a cada evento.

2.2 Arquitectura tecnológica

Para el desarrollo de la plataforma, se ha usado una arquitectura desacoplada que usa React para asistir en el desarrollo del frontend, Django para el backend, donde se implementan todas las variaciones para la generar logs y capturas, y PostgreSQL que se utiliza como base de datos (cf. Fig. 1). Dentro de ésta, cabe destacar dos puntos: (1) frontend y backend se comunican con conexiones HTTP a través de una API REST soportada por Django REST framework. Y (2) considerando que Django permite la modularización de las funcionalidades del sistema, si nos fijamos específicamente en el de experimentos podemos ver que hace un reconocimiento de todos scripts de *python* que están presentes en la carpeta de *plugins*, y carga las funciones presentes en ellos para ser utilizadas en el sistema como funciones de variabilidad. Por tanto para añadir una nueva, basta con almacenarla en la carpeta *plugins*.

2.3 Demostración

En este caso, exhibiremos una demostración de generación de UI logs y capturas en base a un proceso definido. Dicho proceso simula la operación de un operador que comprueba el correo, accede a uno de ellos, y extrae la información adjunta para ser almacenada en un CRM.² Se configuran variaciones a *nivel de caso* sobre: (1) los datos del UI log, en concreto la definición los clics del ratón cuando se accede al correo y el timestamp automático. (2) Y las captura de pantalla, donde primero se modifica el adjunto al correo, que se variará entre un conjunto de DNIs seleccionados, y luego se agrega una dependencia entre capturas de diferente actividad, para que se use el mismo DNI aleatorio a lo largo de un caso específico, para mantener la coherencia entre actividades del mismo caso.

3 Conclusiones

En este artículo se presenta AGOSUIRPA, una plataforma de código abierto que ofrece a la comunidad una forma de generar de forma sintética UI Logs con sus capturas de pantalla asociadas. Estos datos se pueden utilizar para diversos fines en el contexto de RPA, e.g., generación de experimentos de forma sistemática o testeo de robots con el fin de comprobar su correcto funcionamiento antes de ponerlos en producción. Aunque AGOSUIRPA ya es una herramienta funcional, su roadmap conlleva los siguientes trabajos futuros: (1) aumento del número de funciones de variabilidad, para permitir más opciones sobre los datos generados, con datos más realistas y orientados a dominios específicos, (2) mejorar la experiencia de usuario mediante la integración de funciones que faciliten la configuración del UI Log semilla, (3) ampliar las funcionalidades proporcionadas por la herramienta, para aportar metadatos de las imágenes generadas automáticamente, con el objetivo de obtener conjuntos de datos etiquetados para modelos de extracción de características en imagen.

References

1. Albuquerque, G., Lowe, T., Magnor, M.: Synthetic generation of high-dimensional datasets. *IEEE transactions on visualization and computer graphics* **17**(12), 2317–2324 (2011)
2. Enríquez, J.G., Jimenez-Ramirez, A., Dominguez-Mayo, F., Garcia-Garcia, J.: Robotic process automation: a scientific and industrial systematic mapping study. *IEEE Access* **8**, 39113–39129 (2020)
3. Jiménez-Ramírez, A., Chacón-Montero, J., Wojdysky, T., Gonzalez Enriquez, J.: Automated testing in robotic process automation projects. *Journal of Software: Evolution and Process* p. e2259 (2020)
4. Leno, V., Dumas, M., Maggi, F.M., La Rosa, M.: Multi-perspective process model discovery for robotic process automation. In: *CAiSE* (2018)
5. Llewellyn Evans, G.: Disruptive technology and the board: The tip of the iceberg. *Economics and Business Review* **3**(1) (2017)
6. Soto, A., Mora, H., Riascos, J.A.: Web generator: An open-source software for synthetic web-based user interface dataset generation. *SoftwareX* **17**, 100985 (2022)

² Vídeo: <https://youtu.be/RrrNs2wczos>