

PTAC: Una herramienta para testing pasivo de sistemas con comunicaciones asíncronas

M^a Azahara Camacho Magriñán, Inmaculada Medina Bulo y Mercedes G. Merayo

Grupo UCASE de Ingeniería del Software, Departamento de Ingeniería Informática
Universidad de Cádiz, Escuela Superior de Ingeniería, Cádiz, España,
azahara.camacmagri@alum.uca.es, inmaculada.medina@uca.es

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid, Madrid, España,
mgmerayo@fdi.ucm.es

Resumen Las técnicas de pruebas pasivas nos ofrecen la ventaja de poder testear sistemas con los cuales no es posible la interacción directa. Actualmente, la gran mayoría de las propuestas basadas en este concepto asumen que las trazas analizadas reflejan el comportamiento real del sistema, sin tener en cuenta el canal a través del cual han sido observadas. Si la comunicación es asíncrona, la traza observada puede no corresponderse con la producida durante la ejecución del sistema. La necesidad de considerar este tipo de aspectos, llevó a la propuesta de una nueva metodología de pruebas pasivas centrada en canales asíncronos First Input First Output (FIFO). En este artículo se presenta PTAC, una herramienta software que automatiza la aplicación de esta técnica, permitiendo el análisis del comportamiento de los sistemas, mediante la validación de las diferentes propiedades que lo caracterizan a través de trazas reales.

Palabras claves: Prueba de software, pruebas pasivas, sistemas asíncronos.

1. Introducción

Dentro de las diferentes técnicas de validación de software, las técnicas de testing son las más aplicadas en la industria. Existe una clasificación general de dichas técnicas que las divide en pruebas activas y en pruebas pasivas dependiendo de si es o no posible interactuar con el sistema bajo testeo. Gran parte de los sistemas pueden ser evaluados mediante la aplicación de técnicas de pruebas activas, pero existen otros que únicamente pueden ser testeados mediante metodologías pasivas [5].

La mayoría de las metodologías de pruebas pasivas propuestas asumen que las trazas analizadas muestran el comportamiento real del sistema. Sin embargo, es posible que esas trazas hayan sido observadas a través de un canal asíncrono

[6] que haga que la observación del sistema no se corresponda con la ejecución real, ya que algunas acciones pueden aparecer retrasadas en la traza observada con respecto a la traza original.

La necesidad de considerar este tipo de entornos asíncronos y sus peculiaridades llevó a la propuesta de una nueva metodología de pruebas pasivas centrada en canales First Input First Output (FIFO) [4]. Esta propuesta tiene como objetivo la evaluación en tiempo real del cumplimiento, por parte de estos sistemas, de propiedades que describen el comportamiento que deben mostrar.

En este artículo se presenta PTAC (Passive Testing with Asynchronous Communications), una herramienta que implementa dicha técnica. Específicamente, permite el análisis del comportamiento de los sistemas considerados, respecto a diferentes propiedades establecidas previamente.

El resto del artículo presenta la siguiente estructura. En la Sección 2 se introducen los fundamentos teóricos de la técnica implementada. La Sección 3 describe las funcionalidades de la herramienta desarrollada, así como el caso de estudio aplicado para su evaluación. Finalmente, en la Sección 4 se presentan las conclusiones y algunas líneas de trabajo futuro.

2. Metodología de la Técnica

La novedad que aporta la técnica propuesta en [4] es el tratamiento y consideración de las diferencias existentes entre el comportamiento real de un sistema y el percibido por el testeador a través de un canal asíncrono. La principal característica que reflejan estas observaciones son retrasos temporales que provocan una alteración en la secuencia de las acciones sucedidas durante la ejecución del sistema.

La técnica en la que se basa PTAC se centra en un algoritmo que genera un autómata que aceptará las secuencias observadas en un sistema asíncrono, que no satisfagan una propiedad previamente definida. Las propiedades consideradas en el marco propuesto indican las acciones válidas que pueden producirse en el sistema, si este ha generado una traza determinada.

La construcción del autómata final se realiza en dos fases:

- Se genera un grafo inicial con todas las secuencias de acciones que pueden observarse si el sistema ha producido la traza considerada en la propiedad.
- El autómata es adaptado para considerar todas las posibles acciones que pueden haberse producido antes y después de generarse la traza en el sistema, incluyendo aquellas que puedan darse debido a la naturaleza FIFO del entorno.

Si la traza observada se ha generado debido a la producción de una secuencia de acciones por parte del sistema que no cumple la propiedad asociada al autómata construido, dicha traza será aceptada por el autómata indicando que no se ajusta a los requisitos presentados. En el caso de que se ajuste a la propiedad o su observación sea una combinación de acciones compatible con dicha propiedad, se continuará la evaluación de la traza.

3. Descripción de la herramienta PTAC

PTAC es una aplicación que incluye una interfaz gráfica desarrollada en Java con la ayuda de los componentes ofrecidos por SWT [3] y se suministra de la API de Java en código abierto de GraphViz para poder representar los gráficos generados a partir de la aplicación del algoritmo con el lenguaje DOT [1]. Las principales funcionalidades que ofrece el sistema son:

- Gestión de propiedades: Permite al usuario la definición, modificación y eliminación de las propiedades de un sistema.
- Generación de autómatas: Mediante la aplicación del algoritmo descrito en la Sección 2, se obtiene el grafo correspondiente al autómata asociado a la propiedad considerada.
- Sistema de testeo: Permite al usuario comprobar la validez de trazas observadas frente a una propiedad específica.

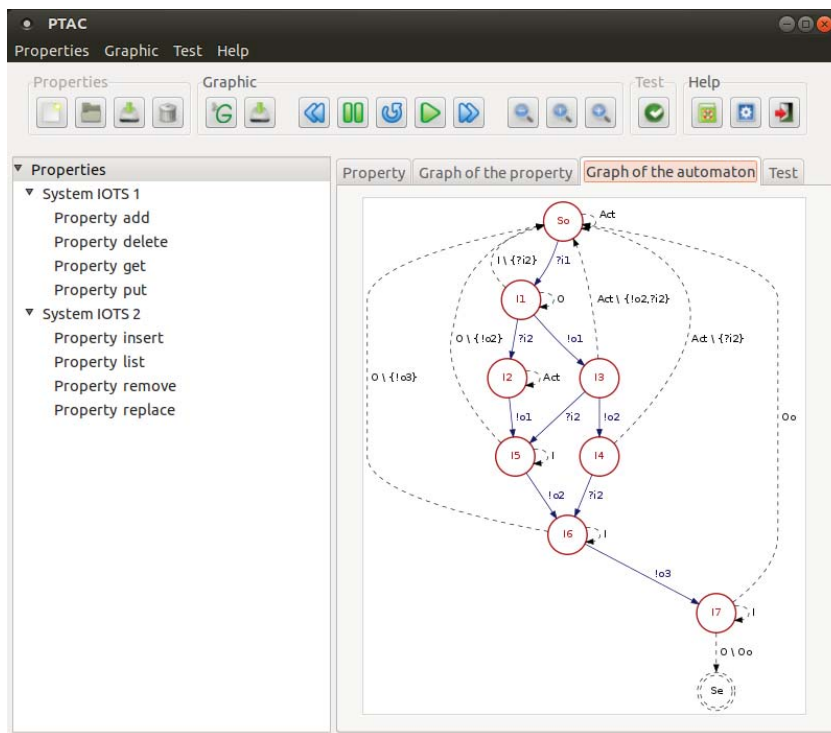


Fig. 1: Muestra de la generación de un autómata con PTAC

Esta aplicación ha sido desarrollada para su ejecución en cualquier máquina con procesadores de 32 ó 64 bits y existen versiones para sistemas operativos *Windows* y *Linux*.

PTAC ofrece a los usuarios la posibilidad de llevar a cabo pruebas de un gran nivel de complejidad, sin necesidad de poseer unos amplios conocimientos. Conocer las propiedades del sistema que quiere testear y saber adaptarlas al formato aceptado por PTAC serán los únicos requisitos que deberá cumplir para realizar un proceso de testing adecuado.

Entre las diferentes ventajas que ofrece esta herramienta se encuentran la reducción del coste temporal de dicho proceso gracias a la automatización proporcionada por PTAC, la reducción significativa de posibles errores humanos que pueden cometerse a la hora de llevar a cabo dicha aplicación de forma manual y la difusión de la técnica implementada a través de una interfaz cómoda para el usuario.

Un ejemplo de su utilidad es la aplicación de PTAC en el caso de estudio del protocolo de red HyperText Transfer Protocol (HTTP) [2]. En el experimento se definieron diferentes propiedades que deben cumplirse en las comunicaciones del protocolo y se capturaron trazas reales para comprobarlas. Estas trazas fueron mutadas y sometidas a testeo en PTAC, obteniendo como resultado la detección de los errores inyectados en las trazas.

4. Conclusiones y Trabajo Futuro

En este trabajo se ha presentado PTAC, una aplicación que incluye una interfaz gráfica para realizar testing pasivo en base a la propuesta presentada en [4], para sistemas que presentan asincronía en los canales de observación.

Como trabajo futuro, se pretende continuar con el desarrollo y perfeccionamiento de cada una de las funcionalidades que posee PTAC, con la intención de hacer lo más manejable posible el proceso de testing. También se ha planteado la posibilidad de crear un modo estudiante incluido dentro de la misma interfaz, más detallado a la hora de mostrar resultados y procesos intermedios, haciendo posible la divulgación del testing de software entre los estudiantes de Informática.

Referencias

1. DOT Manual, <http://www.graphviz.org/cgi-bin/man?dot>
2. Especificación del protocolo HTTP. <http://www.ietf.org/rfc/rfc2616.txt>
3. SWT, The Standard Widget Toolkit, <http://www.eclipse.org/swt/>
4. Hierons, R.M., Merayo, M.G., Núñez, M.: Passive testing with asynchronous communications. In: IFIP 33rd Int. Conf. on Formal Techniques for Distributed Systems, FMOODS/FORTE'13, LNCS 7892. pp. 99–113. Springer (2013)
5. Lee, D., Netravali, A., Sabnani, K., Sugla, B., John, A.: Passive testing and applications to network management. In: 5th IEEE Int. Conf. on Network Protocols, ICNP'97. pp. 113–122. IEEE Computer Society (1997)
6. Núñez, M.: Formal testing of timed and probabilistic systems. In: 23rd Int. Conf. on Testing Software and Systems, ICTSS'11, LNCS 7019. pp. 9–14. Springer (2011)