



UNIVERSIDAD COMPLUTENSE MADRID

Proyecto de Innovación y Mejora de la Calidad Docente

Convocatoria 2015

Proyecto nº 281

Aplicación del sistema jPET para la generación automática de tests en asignaturas de programación con Java

Responsable del proyecto: Miguel Gómez-Zamalloa Gil

Facultad de Informática

Departamento de Sistemas Informáticos y Computación

1. Objetivos propuestos en la presentación del proyecto

El testeo o "testing" de programas es la técnica más utilizada para razonar acerca de la corrección de programas informáticos. Es bien sabido que su coste es muy elevado - se estima que más del 50% de los costes totales en el desarrollo de software se dedican al testing. Básicamente, el testing de un programa consiste en ejecutar dicho programa con una serie de casos de prueba o tests, generados manualmente, con el objetivo de comprobar que el resultado es el esperado. En el ámbito de la enseñanza de la programación a nivel universitario, el testing de programas juega un papel diferente aunque también fundamental en dos contextos distintos. Por un lado, los estudiantes de programación deben emplear mucho tiempo realizando pruebas para asegurarse de que sus trabajos de programación, bien sean ejercicios, prácticas, proyectos o incluso exámenes, funcionan como se espera. Por otro lado, los profesores se ven obligados a realizar pruebas, habitualmente de manera manual y una por una, de cada trabajo sometido a corrección. Existen sistemas que ayudan en la fase de testing de diferentes maneras. Por ejemplo, para el lenguaje de programación Java, el sistema JUnit proporciona una sintaxis en la que escribir los casos de prueba, y un mecanismo para que estos sean ejecutados de manera automática, pudiendo observarse qué casos producen un comportamiento correcto o incorrecto. El esfuerzo del testing por tanto se centra fundamentalmente en la definición de los casos de prueba. Existen multitud de métricas, llamadas criterios de adecuación, que evalúan la calidad de un conjunto de casos de prueba de acuerdo a diferentes características, con el objetivo de poder medir así la calidad de la fase de testing. No es en absoluto sencillo generar casos de prueba que satisfagan los criterios más sofisticados.

Por otro lado, en la actualidad, la mayoría de los lenguajes de programación que se utilizan en los distintos grados de la Facultad de Informática están basados en el paradigma de objetos, y más concretamente en el lenguaje Java. En particular, para este lenguaje existe una biblioteca de programas denominada JUnit para comprobar la corrección parcial de un programa suministrando casos de prueba. De esta forma, si el programa ejecuta correctamente dichos casos de prueba, este puede considerarse correcto para dichos tests. JUnit es el estándar industrial de facto para la realización de pruebas unitarias en Java, lo que lo hace especialmente relevante en la docencia de las asignaturas de programación. El problema fundamental del uso de JUnit estriba en que la confección de los casos de prueba es manual, lo que conlleva que el esfuerzo en generarlos sea en muchos casos mayor que el esfuerzo que supone programar la propia aplicación.

El grupo de investigación al que pertenecen los integrantes de esta propuesta, ha trabajado en los últimos años en la generación automática de casos de prueba utilizando técnicas formales, desarrollando técnicas que permiten obtener de manera totalmente automática casos de prueba que satisfagan determinados criterios de

adecuación. En particular, desde hace varios años se viene desarrollando el sistema jPET, un generador de casos de prueba para programas Java, que garantiza la satisfacción de una serie de criterios de adecuación para los tests obtenidos. Se puede encontrar más información en la web <http://costa.ls.fi.upm.es/pet> y hacerse una idea más detallada del funcionamiento de la herramienta en el vídeo disponible en http://costa.ls.fi.upm.es/pet/demo_video.html

El objetivo de este proyecto de innovación ha sido estudiar la aplicación, adaptación e implantación del sistema jPET, para su uso tanto por el alumno como por el profesor, en las asignaturas de programación con el lenguaje Java de los grados de la Facultad de Informática. Esto ha conllevado, entre otras cosas: el estudio del uso de jPET con el tipo de programas que se realizan, el desarrollo de nuevas extensiones en el motor de la herramienta para poder tratar con funcionalidades del lenguaje no tratadas hasta ahora, la generación automática de código junit, y, el desarrollo de una interfaz web amigable que permita un uso sencillo y colaborativo de la herramienta a través de internet sin necesidad de realizar instalación alguna.

Es importante hacer notar que varios de los profesores integrantes en este proyecto también imparten clase en las asignaturas en cuestión, lo que proporciona el entorno perfecto para que dicha implantación sea posible.

2. Objetivos alcanzados

Como se ha mencionado en la descripción del documento, el objetivo fundamental de este proyecto de innovación es estudiar la aplicación, adaptación e implantación del sistema jPET, para su uso tanto por el alumno como por el profesor, en las asignaturas de programación con el lenguaje Java de los grados de la Facultad de Informática. Las asignaturas susceptibles de ser consideradas son fundamentalmente Tecnología de la Programación, Ingeniería del Software (ambas asignaturas de 2º de los grados) y Métodos Algorítmicos en Resolución de Problemas (asignatura de 3º del grado en Ingeniería Informática) y sus diferentes variantes en el resto de grados. Esto permitiría, por un lado, dotar a los estudiantes de una herramienta para ayudarles a razonar sobre la corrección de sus programas, y en definitiva a poder realizar trabajos de mayor calidad; y por otro lado, dotar a los profesores de una herramienta que les ayudaría en los procesos de corrección de ejercicios, prácticas y exámenes haciendo que este proceso de corrección fuese más fácil, y sobretodo más fiable.

La herramienta está actualmente implementada para programas escritos en el lenguaje de programación Java, quizás el lenguaje más utilizado a nivel mundial y el lenguaje elegido en la mayoría de las asignaturas de programación y por la mayoría de estudiantes para la realización de sus proyectos fin de carrera, etc. Cabe destacar que el motor de la herramienta es, en gran parte, independiente del lenguaje de programación y por tanto sería factible su reutilización para poder tratar con otros lenguajes de programación, como por ejemplo C++, lenguaje utilizado en las asignaturas de iniciación a la programación en el primer curso de los grados, o Python, un lenguaje emergente y utilizado en la asignatura de iniciación a la programación en el grado de Matemáticas.

Por otro lado, las asignaturas de programación consideradas en el proyecto, vienen utilizando en los últimos años el llamado “juez online”. Estos jueces online originalmente se han utilizado en concursos de programación. Permiten por tanto por un lado publicar problemas de programación, y en una segunda fase, que los concursantes registrados suban sus soluciones, las cuales son evaluadas en el acto produciendo un veredicto de corrección. Además, registran estadísticas de distintos aspectos de uso de los distintos concursantes. El uso de estos jueces online está resultando muy práctico como medio para que los alumnos puedan realizar y autoevaluar ejercicios prácticos de programación, y también para los profesores como medio de evaluación añadido a los exámenes finales. Para que la autoevaluación realizada por estos jueces sea efectiva, los profesores deben proporcionar casos de prueba de calidad, los cuales son ejecutados automáticamente cuando los alumnos suben sus soluciones. Escribir estos casos de prueba resulta costoso y complejo, y es en este punto, donde el uso de jPET podría resultar muy útil. Los casos de prueba generados por jPET podrían servir como punto de partida a la hora de generar un conjunto de casos de prueba de calidad. No obstante, sería razonable que los

profesores complementasen estos casos de prueba con casos que evalúen aspectos más específicos de las soluciones que se exigen, como la complejidad computacional de estas.

De cara a abordar dicho objetivo fundamental se han identificado y realizado las siguientes tareas:

1) Se ha realizado un estudio del uso de jPET con el tipo de programas que se realizan en las asignaturas consideradas. En particular, se han considerado los programas que los distintos profesores de las asignaturas en cuestión han ido publicando en los jueces online. Se concluye que el uso de jPET es factible para la mayoría de estos programas, excepto quizás para los problemas más avanzados planteado en las asignaturas del tercer curso. No obstante, ha sido necesario el desarrollo de nuevas extensiones en el motor de la herramienta para poder tratar con funcionalidades del lenguaje Java no tratadas anteriormente.

2) Se ha implementado una extensión en el sistema jPET que permite la generación automática de código junit. De esta manera los tests generados pueden ser directamente ejecutados en el entorno y todo el proceso puede ser fundamentalmente automático.

3) Se ha implementado una extensión en jPET que traduce los casos de prueba generados al formato de texto aceptado por los jueces online.

4) Se ha estudiado la integración de jPET dentro del juez online DomJudge (<https://www.domjudge.org>). Dicha integración, aunque factible, resulta compleja, y se encuentra aún en fase de desarrollo.

5) Se ha trabajado, y se plantea seguir trabajando, en el desarrollo de una interfaz web amigable que permita un uso sencillo de jPET y que permita aprovechar todas las funcionalidades de la herramienta a través de internet sin necesidad de realizar instalación alguna, y idealmente permitiendo guardar y compartir sesiones y/o proyectos.

3. Metodología empleada en el proyecto

Se han identificado una serie de tareas orientadas a conseguir los objetivos mencionados. Cada una de estas tareas ha tenido un responsable principal, que ha diseñado el plan para conseguir el objetivo de la tarea, delegando para ello en los demás miembros del proyecto de la manera que ha considerado oportuna. La metodología utilizada durante el proceso de diseño e implementación ha sido la habitual de los proyectos de desarrollo de aplicaciones (ver más detalles en el apartado 5).

En el informe final, el responsable de cada tarea se ha encargado de la parte del informe correspondiente a su tarea, de nuevo posiblemente delegando a su conveniencia. El responsable principal del proyecto ha sido el encargado de juntar las distintas partes del informe y producir el informe final.

4. Recursos humanos

Todos los profesores integrantes del proyecto tienen amplia experiencia en la impartición de asignaturas relacionadas con la programación, y en particular con el lenguaje Java. La mayoría de los profesores son además desde hace años profesores responsables en las asignaturas en las que se va a estudiar la implantación del sistema jPET. Por otro lado, los fundamentos técnicos e implementación de la herramienta están basados en la investigación continuada de los miembros del proyecto en testing formal y análisis estático.

El grupo de trabajo se ha compuesto de tal manera que todos los aspectos del proyecto queden cubiertos por los conocimientos de sus miembros. Más concretamente:

Miguel Gómez-Zamalloa Gil: Lleva 10 años impartiendo asignaturas relacionadas con la programación. Su investigación se ha centrado en el testing formal basado en la ejecución simbólica (tecnología de base de la herramienta objeto de este proyecto). Es el principal desarrollador, junto con Elvira Albert, de la herramienta jPET.

Purificación Arenas Sánchez: Tiene una dilatada experiencia docente de casi 25 años. Es desde hace años profesora responsable de "Tecnología de la Programación", asignatura de programación de los nuevos grados, impartida en el lenguaje Java y principal asignatura objeto de la implantación del sistema jPET.

Elvira Albert Albiol: Lleva 15 años impartiendo clases relacionadas con la programación. Elvira lidera el grupo de investigación COSTA, y además, junto con Miguel Gómez Zamalloa, es creadora y responsable de la herramienta jPET.

Jesús Correas Fernández: Tiene 12 años de experiencia docente. En la actualidad imparte la asignatura "Tecnología de la programación".

Samir Genaim: Tiene una amplia experiencia impartiendo este tipo de asignaturas en diferentes universidades y en la actualidad imparte también "Tecnología de la programación". Parte de su investigación en los últimos años está centrada en el desarrollo de un entorno web colaborativo y genérico para el uso de herramientas de análisis de programas y será por tanto pieza fundamental para integrar jPET en dicho entorno web.

Guillermo Román Díez: Profesor Asociado 4+4 en la Univ. Politécnica de Madrid, impartiendo "Programación para Sistemas" y "Concurrencia". Esta última asignatura está centrada en los aspectos concurrentes del lenguaje Java. Así mismo, ha impartido las asignaturas "Programación II" y "Estructuras de Datos II", ambas impartidas en el lenguaje Java. Además, es experto en el desarrollo de aplicaciones en el lenguaje Java y aplicaciones web, debido a su experiencia profesional (6 años) como Analista y Jefe de Proyecto en diversas empresas en el sector TIC.

5. Desarrollo de las actividades

A continuación se enumeran las tareas o actividades que se han llevado a cabo orientadas a conseguir los objetivos mencionados. Cada una de estas tareas ha tenido un responsable principal, quién ha diseñado el plan para conseguir el objetivo de la tarea, habiendo delegado para su realización en unos colaboradores de la manera que el responsable ha considerado oportuna.

Tarea 1. Estudio de aplicabilidad de jPET por asignatura: Se ha analizado la aplicabilidad de la herramienta jPET en cada asignatura de programación que utiliza el lenguaje Java. Esto ha requerido examinar los correspondientes temarios, prácticas y ejercicios, así como reunirse con otros profesores involucrados en las asignaturas en cuestión. Tras el análisis se concluye que, a parte del posible uso comentado en asignaturas que utilizan el lenguaje Java, jPET podría ser especialmente útil en el contexto de los llamados “jueces online”. Como se ha comentado en el punto 2 de esta memoria, estos jueces online permiten por un lado publicar problemas de programación, y en una segunda fase, que los alumnos suban sus soluciones, las cuales son evaluadas en el acto produciendo un veredicto de corrección. El uso de estos jueces online está resultando muy práctico como medio para que los alumnos puedan autoevaluar ejercicios prácticos de programación, y también para los profesores como medio de evaluación añadido a los exámenes finales. Para que la autoevaluación realizada por estos jueces sea efectiva, los profesores deben proporcionar casos de prueba de calidad que son ejecutados automáticamente cuando los alumnos suben sus soluciones. Escribir estos casos de prueba resulta costoso y complejo, y es en este punto, donde el uso de jPET podría resultar muy útil. Los casos de prueba generados por jPET podrían servir como punto de partida a la hora de generar un conjunto de casos de prueba de calidad. Un aspecto muy interesante en este sentido, es que no es necesario que los programas de los alumnos utilicen el lenguaje Java. Lo único que sería necesario es que el profesor proporcione una solución escrita en Java. A partir de esta se podrían generar los casos de prueba iniciales. Si se diese el caso de que las soluciones de los alumnos viniesen escritas en Java, se podrían plantear enfoques más interesantes en los cuales los tests se forman a partir de ambas soluciones, la del alumno, para generar los datos de entrada, y la del profesor, para chequear que las salidas para esas entradas son las correctas.

Tarea 2. Recopilación de programas: Se han recopilado ejemplos de programas (ejercicios, ejemplos y prácticas) realizados en las asignaturas en cuestión en los últimos dos cursos. Muchos de los problemas se han obtenido desde los propios jueces online instalados en cada asignatura. Esto permitirá en el futuro poder realizar estudios de calidad de los tests obtenidos con jPET respecto a los ya proporcionados por los profesores, y lo que es más interesante, obtener resultados experimentales de

la calidad de los tests, en base a medir los errores que éstos exponen sobre los programas ya subidos por los alumnos.

Tarea 3. Pruebas y depuración en jPET: Se han ido realizando pruebas con jPET utilizando los programas recopilados en la tarea anterior con el objetivo de establecer los límites de uso de la herramienta y saber en qué contextos puede ser viable su uso y en cuáles no. En el proceso se han corregido algunos errores en el funcionamiento de jPET y se han ido anotando futuras extensiones necesarias para tratar con características no soportadas por el sistema en su estado actual.

Tarea 4. Extensiones en jPET: Basándonos en la información obtenida en la tarea anterior, se han estudiado y desarrollado las extensiones que, siendo viables, han permitido ampliar la funcionalidad de jPET para que sea capaz de tratar con un espectro más amplio de programas. En particular, se ha extendido el número de librerías del lenguaje Java soportadas por jPET. En este sentido, cabe destacar el tratamiento de las operaciones con cadenas de caracteres, el cual es muy habitual en estos programas, y por el momento ha habido que reescribir ligeramente los programas para poder utilizarlos. La implementación de esta extensión en jPET ya se ha realizado parcialmente y se espera esté lista en un futuro cercano, lo que permitirá tratar muchos de los programas directamente, sin necesidad de intervención por parte del usuario.

Tarea 5. Generación automática de junit: Se ha implementado el soporte en jPET para la generación automática de código junit para los tests obtenidos. Aunque ya existía un soporte parcial en este sentido en la herramienta, éste está actualmente dentro del plugin de Eclipse y tenía que ser replanteado y reimplementado para poder hacer uso de él desde la interfaz web mencionada en la siguiente tarea. Esta funcionalidad está aún en proceso de implementación y se espera que esté funcionando en un futuro cercano.

Tarea 6. Desarrollo de una interfaz web: Aunque ya existía una interfaz web para jPET, ésta no funcionaba correctamente, no permitía aprovechar muchas de las funcionalidades de jPET, y, además no estaba pensada para poder ser utilizada por alumnos ni profesores. Se ha planteado el desarrollo de una interfaz web amigable, totalmente nueva, pensada para poder ser utilizada por profesores y alumnos de manera que éstos puedan utilizar todas las funcionalidades de jPET de manera sencilla. Aunque gran parte de dicha interfaz ya se ha desarrollado y probado con una herramienta similar más sencilla, la integración de la interfaz con jPET se encuentra aún en fase de desarrollo al haber surgido diversas complicaciones. Se espera tener la interfaz web totalmente integrada con jPET en un futuro cercano. Además, se espera poder disponer de un soporte para registrar usuarios y así poder guardar y compartir proyectos, proporcionando un entorno colaborativo.