

Usabilidad & AOP: desarrollo y evaluación de un framework de dominio

Sandra Casas, Natalia Trejo y Roberto Farías

GISP - Instituto de tecnología Aplicada
Universidad Nacional de la Patagonia Austral
Rio Gallegos, Av. Gregores y Piloto Lero Rivera S/N
scasas@unpa.edu.ar

Resumen

La usabilidad [1] es una disciplina que procura la mejora en el uso del software. Se han propuesto diversos métodos para su evaluación e incluso la combinación de ellos, en esencia se basan en el cálculo de métricas respecto de cierto factor de usabilidad (efectividad, eficiencia y satisfacción). En todos los casos una cuestión crítica en el proceso de evaluación, es la captura o recolección de datos que tiene impacto directo en el análisis y evaluación de la usabilidad. Sucede que las aplicaciones software actuales, son cada vez más complejas y heterogéneas (web-móviles-escritorio), en sus características (arquitectura-diseño-implementación) por lo que se requieren métodos de recolección de datos automáticos más efectivos y especializados que los simples archivos de logs generados por los servidores o sistemas operativos. Se observa como un requerimiento la necesidad de aplicar enfoques que permitan extraer datos más específicos y significativos con este tipo de aplicaciones, que a la vez no sean invasivos en las aplicaciones y sean lo suficientemente flexibles para ser reutilizados en aplicaciones de contextos similares. Con el objeto de plantear una solución a la captura de datos automática para la evaluación de la usabilidad de características flexible, configurable, reutilizable y no intrusiva, este proyecto propone des-

arrollar un framework de dominio que facilite la reutilización del diseño e implementación de aspectos dando soporte a todas las funcionalidades automatizables que la usabilidad conlleva.

Palabras clave: Usabilidad, AOP, AspectcJ, Framework, Reutilización.

Contexto

El grupo de i+d GISP ha desarrollado diferentes actividades relacionadas al área de Separación de Concerns, más específicamente al Desarrollo de Software Orientado a Aspectos desde el año 2005. Las líneas de abordaje se podrían clasificar según sus objetivos en dos principalmente: el estudio de características y mecanismos de los enfoques a efectos de proponer mejoras y su aplicación a contextos/dominios no explorados. Así se ha trabajado en las siguientes temáticas: Conflictos entre aspectos, Aspectos Tempranos, Minería de Aspectos y Refactoring Aspectual, Reglas de Negocio, Mantenimiento y evolución del software. Se han ejecutado seis proyectos de investigación acreditados en la UNPA. La producción del GISP está disponible en <http://sites.google.com/site/profeprog/>

Introducción

Las cuatro formas básicas de evaluación son: automática (se calculan las métricas durante la ejecución de la aplicación), empírica (la usabilidad es evaluada testeando la aplicación con usuarios reales), formal (usando modelos formales y fórmulas para el cálculo de medidas de usabilidad), e informal (basados en reglas generales y la habilidad y experiencia de los evaluadores). Este proceso de evaluación implica varias actividades en función del método empleado, pero en general se agrupan en 3 pasos básicos [2]: Captura de datos, Análisis y Crítica. Las pruebas de usabilidad se pueden realizar en diferentes etapas del desarrollo del software. Algunas técnicas de evaluación, sólo se pueden aplicar cuando existe un prototipo de la aplicación, mientras que otras se pueden aplicar a principios de la etapa de diseño. Cada técnica tiene sus propios requerimientos y generalmente diferentes técnicas descubren diferentes problemas. También depende con el tipo de software y desarrollo, por ejemplo el software libre/código abierto, seguramente la evaluación de usabilidad será cuando esté finalizado o sea adoptado. La evaluación de usabilidad puede ser empleada para adaptar software a nuevos contextos, mercados, y hacerlo más competitivo.

Limitaciones de la captura de datos automática.

La recolección de datos puede hacerse a través de diversos métodos, como ya se ha mencionado. En lo que respecta a la captura de datos automatizada se ha caracterizado por usar archivos de logs generados por servidores web o sistemas operativos. El uso de logs o archivos de logs es una de las técnicas de captura de datos más usada para obtener información de usabilidad. Sus ventajas son varias, por ejemplo: a) soportan la ejecución de las pruebas en laboratorios de usabilidad o en forma remota, en el ambiente natural del

usuario; b) son transparentes para los usuarios; c) permiten con facilidad obtener información de múltiples usuarios; d) permiten detectar errores actuales; etc. Los logs aportan gran cantidad de datos a los evaluadores de usabilidad, además de poder contrastar comportamientos entre diferentes usuarios. Sin embargo, en la actualidad los logs tienen desventajas que han originado diversas discusiones. Las aplicaciones actuales requieren el análisis de patrones de interacción mucho más complejos, no sólo las rutas de navegación. Puede ser necesario registrar variables de situación como la cantidad de información presentada, y requerida al usuario. Además se puede requerir información específica de la aplicación que solo puede ser obtenida del análisis dinámico de la aplicación. Es decir, los logs estándar que generan los servidores web ya no son suficientes ya que no pueden ofrecer información específica y detallada. Por lo tanto los desarrolladores necesitan diseñar e integrar mecanismos de registros adecuados “directamente” a las aplicaciones web. Por otro lado, debido a la gran cantidad de datos que generan los logs y que estos no están pensados para posteriormente ser usados para la evaluación de la usabilidad, es que la tarea de análisis de archivos de registro también creció en complejidad. En estos casos, es necesario aplicar otras técnicas, por ejemplo del área de minería de datos, para que los evaluadores de usabilidad puedan comprender e interpretar esa información. Así y todo ocurre que el análisis de logs no responde a todas las preguntas de usabilidad y en algunos casos probablemente plantee nuevos interrogantes. Dado que muchas veces no se conoce el contexto de uso, queda preguntarse por qué un usuario se comportó de una cierta forma, probablemente inesperada. En otras palabras, la interpretación de los datos puede ser re-

almente problemática y propensa a errores.

AOP/USABILIDAD

AOSD/AOP [3] [4] han sido reconocidas como técnicas validas para superar las dificultades señaladas. Un conjunto de incipientes y recientes trabajos dan cuenta de estas experiencias:

En [5] se propone un enfoque para el diseño de herramientas que soporten la evaluación automática de la usabilidad usando AOP. Se analiza que mediante una jerarquía de aspectos se pueden reusar pointcuts que compartan los mismos puntos de entrada y salida (join-points). Sin embargo la estructura es específica de cada aplicación en la que se use. Se presentan simples diagramas y código de ejemplo. En [6] se presentan y comparan dos enfoques para la evaluación temprana de la usabilidad en aplicaciones interactivas usando trazas. El primer enfoque reside en emplear AOP y el segundo agentes interactivos. Ambos enfoques requieren ser incluidos en los estudios preliminares y de factibilidad del proyecto, especificación y codificación específica. AOP no requiere diseño arquitectónico, a diferencias de los agentes dado que son parte de tales estructuras. Es decir AOP permite mantener intacto el código inicial. En [7] se propone AOP para capturar automáticamente eventos de la interface de usuario. Se propone una jerarquía de aspectos, en la cual se logra reusar solo el código relacionado a la métrica (hora-fecha y evento ocurrido), los pointcuts y advices deben ser redefinidos en cada caso y aplicación. Se presentan códigos de ejemplo y un caso de estudio simple. En [8] se propone la implementación de aspectos para capturar algunos eventos de interface. Se plantea usar Objective-C, agregando los aspectos a la jerarquía de objetos usando la tecnología “method swizzling”, e extendiendo de clases. No se presentan

códigos, ni diagramas del diseño o implementación. En [9] se propone el uso de aspectos para la captura automática de datos de interacción de usuario. El modelo esencial se propone la generación automática de los aspectos. A través de la definición de tareas en formularios/XML y una biblioteca de métricas de usabilidad. Se presenta una herramienta y un caso de estudio sobre una aplicación Androide.

En [10] se presenta una metodología y un prototipo funcional de un toolkit que consiste en un framework para aplicaciones móviles, un asistente IDE y el servidor web, que permite a los desarrolladores realizar experimentos relacionados al campo de la usabilidad. Con el enfoque presentado son capaces de conectarse de forma transparente a los datos de interacción de usuario y calcular métricas de usabilidad y datos estadísticos descriptivos durante los estudios de campo sin supervisión a los usuarios finales. Estos datos son utilizados luego para presentar los resultados a los desarrolladores de forma automática. Se adopta AOP para la captura de los datos. En [11] se presenta un enfoque basado en técnicas OA para trazas de eventos de interface de usuario y recolectar información contextual, en aplicaciones de escritorio (WIMP). Propone una jerarquía de aspectos, la cual consiste básicamente reside en definir un subaspectos por tipo de evento. Se presentan simples códigos de ejemplo. En [12] se presenta una metodología para captura datos de usabilidad automáticamente usando aspectos en aplicaciones móviles, además de métricas convencionales se recolecta información del contexto aportada por sensores. La generación de los aspectos es automática a través de una herramienta y librería de métricas. Se presenta el flujo de trabajo para la generación de aspectos.

Líneas de Investigación, Desarrollo e Innovación

Como punto de partida se observa:

a) La heterogeneidad y complejidad del software actual, va mucho más allá de la navegación y presentación de información (sitios web), exigen métodos de captura de datos automáticos específicos. La automatización de la recolección de esta información con técnicas tradicionales (OO-estructurado) conlleva necesariamente a agregar cientos de líneas de código en las aplicaciones cuya funcionalidad no está relacionada a la lógica de negocios o nuevos requerimientos. Por ende, se ensucia e invade la aplicación, se altera el diseño original, además este código agregado presenta bajísimas posibilidades de reuso y su posterior remoción puede ser compleja y ocasionar más errores.

b) La AOSD/AOP ha sido extensamente aplicada a problemas tales como persistencia, tracing, registro de actividades/auditoría, logging, seguridad, interfaz gráfica, etc., sin embargo, son escasos y muy recientes los antecedentes en los cuales se enfoca a la usabilidad como un crosscutting concerns (CCC) o requerimiento no funcional. No obstante, al analizar características y requerimientos de la usabilidad se observa con claridad que ésta implica una serie de cuestiones funcionales directamente relacionadas con CCC tales como: analizar sesiones de usuario, calcular los tiempos de realización de las tareas, identificar errores frecuentes, almacenar en un registro de log tiempos de realización de tareas, etc. Se observa como requerimiento inicial un análisis de los sub-concerns que componen a la usabilidad como concern de mayor nivel.

c) Los framework OA también son un tema de investigación de gran actualidad. El desarrollo se centra en procesos iterati-

vos en los cuales la identificación de hot-spots y la aplicación de patrones guían el diseño e implementación. Las experiencias más validadas refieren principalmente a re-construcciones de framework OO cuyos CCC se modularicen con aspectos. Existe un vacío en cuanto a la indagación del desarrollo de framework OA desde el inicio y como principal técnica de modularización.

El planteo que se propone es: la usabilidad se formula a través de la evaluación de atributos y éstos a través de la aplicación de métricas. Sin embargo, se requieren datos diversos según el tipo de aplicación y atributo que se pretenda evaluar, e incluso podrían imponerse restricciones nuevas. La arquitectura y diseño de cada aplicación software objeto de la evaluación también impone restricciones. Ante tanta heterogeneidad, el diseño y configuración de módulos independientes que cumplan con estas responsabilidades se presenta como una actividad que dista de ser trivial (es más que una traza o un simple logging).

Algunas preguntas que conducen esta propuesta son: ¿Cuáles son los sub-concerns que plantean los atributos de usabilidad y sus correspondientes métricas en su automatización? ¿Cómo éstos se pueden organizar y ordenar para ser configurados y adaptados según el sistema al que se apliquen? ¿De qué manera pueden ser implementadas con aspectos para que puedan ser reusados y/o especializados?

Resultados y Objetivos

Hipótesis. Un framework de dominio que aporte los medios y mecanismos para configurar y capturar la recogida de datos automática, de manera transparente y no intrusiva para las aplicaciones proporcionará mayores oportunidades en su reusabilidad, significancia y valor de la información recolectada.

Objetivo General

Construir un framework basado en aspectos para la captura y recolección de datos automática útiles para las evaluaciones de usabilidad de aplicaciones software.

Objetivos Específicos

- (a) Identificar, conceptualizar y caracterizar los sub-concerns relacionados al concern Usabilidad.
- (b) Identificar los principales componentes del framework.
- (c) Diseñar e implementar el framework aspectual.
- (d) Evaluar el framework.

Formación de Recursos Humanos

En particular este proyecto incluye 2 tesis de la Maestría en Informática y Sistemas (UNPA) y 1 becario de postgrado.

Referencias

- [1] ISO 9241-11. Ergonomic requirements for office work with visual display terminals. ISO, 1998.
- [2] M. Ivory y M. Hearst, "The State of the Art in Automating Usability Evaluation of User Interfaces", ACM Computing Surveys, Vol. 33, No. 4, December 2001.
- [3] Aspect-Oriented Software Development, 2004. <http://aosd.net>
- [4] Kiczales G., Lamping J., Mendhekar A., Maeda C., Lopes C., Loingtier J., Irwin J. (1997) Aspect-oriented Programming, Proceedings of the European Conference on Object-Oriented Programming.
- [5] Tarta A., Moldovan G. (2006) "Automatic Usability Evaluation Using AOP" Automation, Quality and Testing, Robotics, IEEE International Conference (Volume:2)
- [6] Tarby J., Ezzedine H., Rouillard J., Tran C., Laporte P., Kolski C. (2007). Traces Us-

ing Aspect Oriented Programming and Interactive Agent-Based Architecture for Early Usability Evaluation: Basic Principles and Comparison by: In HCI (1), Vol. 4550, pp. 632-641

[7] Yonglei T., (2008) "Automated Data Collection for Usability Evaluation in Early Stages of Application Development" 7th WSEAS in ACACOS 08, China.

[8] Holzinger, A., Brugger, M., Slany, W. (2011) Applying Aspect Oriented Programming in Usability Engineering Processes - On the Example of Tracking Usage Information for Remote Usability Testing. Proceeding of the 8th. International Conference on electronic Business and Telecommunications. España pp 53-56

[9] Kronbauer A., Santos C. (2011) "Um modelo de avaliação da usabilidade baseado na captura automática de dados de interação do usuário em ambientes reais"., Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction - Pages 114-123

[10] Lettner F., Holzmann C.: Automated and unsupervised user interaction logging as basis for usability evaluation of mobile applications. (2012) The 10th International Conference on Advances in Mobile Computing & Multimedia, Indonesia 118-127

[11] Yonglei T. (2012) "Aspect-Oriented Instrumentation for Capturing Task-Based Event Traces" Int. J. on Control System and Instrumentation, Vol. 03, N0. 01.

[12] Kronbauer A., Santos C., Vieira V., (2012) Um Estudo Experimental de Avaliação da Experiência dos Usuários de Aplicativos Móveis a partir da Captura Automática dos Dados Contextuais e de Interação. Brazilian Symposium on Human Factors in Computing Systems. Brazil