

Experiencia de aplicación de patrones de usabilidad en el diseño de software

Fernandez, Ezequiel Enrique; Solis Luna, Santiago Jose; Fernandez Taurant, Juan Pablo; Perez, Nicolas; Moreno, Juan Carlos; Marciszack, Marcelo Martin

*Centro de investigación, desarrollo y transferencia de Sistemas de Información
Universidad Tecnológica Nacional, Facultad Regional Córdoba
Maestro López esq. Cruz Roja Argentina, Ciudad Universitaria, C.P.: 5016, Córdoba, Argentina.*

Abstract

Hoy en día, en la construcción de una aplicación de software se deben considerar no solamente los Requisitos Funcionales sino también los Requisitos No Funcionales, para desarrollar una aplicación de calidad. Se ha observado, muchas veces, que no se tienen en cuenta los requerimientos no funcionales, sobre todo los de usabilidad, en el momento del diseño, sino más bien cuando el producto se encuentra casi terminado, lo que provoca grandes inconvenientes y costos al proyecto. Por esta razón, se buscó a través de los patrones, como buenas prácticas probadas como soluciones a problemas concretos, verificar si es factible incorporar aspectos de usabilidad en forma temprana en la construcción del software evaluando algunos casos concretos. Asimismo, a través de esta experiencia se busca relacionar los patrones de usabilidad con los patrones de diseño, mediante una experiencia de aplicación de patrón concreto.

Palabras Clave

Patrones de diseño, patrones de usabilidad.

Introducción

Cuando se diseña un software, solo se consideran los requerimientos no funcionales a la hora de definir la arquitectura, por lo que muchos atributos de calidad no son tratados en detalle en etapas tempranas del ciclo de desarrollo.

Uno de los atributos de calidad del software más reconocido e importante, es la usabilidad. Se conoce, que muchos sistemas han fracasado por no tener resueltos aspectos fundamentales de usabilidad.

La usabilidad se debe tener presente en todos los diseños de los sistemas y por lo tanto no debería resolverse a posterior en la

etapa la implementación. Si la usabilidad no se tiene en cuenta tempranamente, previo a la implementación de un sistema, esto puede aumentar la complejidad y los costos de mantenimiento [1].

Este trabajo busca verificar la aplicación de buenas prácticas relacionadas con la usabilidad, incluyendo patrones de Usabilidad durante el diseño, ya que consideramos que sería el momento indicado para abordar los criterios de usabilidad y no cuando el producto está terminado. Esto permitiría relacionar patrones de diseño juntos con los de usabilidad, para potenciar la calidad del sistema.

Desarrollo

Para el desarrollo de este trabajo se ha elegido uno de los criterios de usabilidad más conocido, la posibilidad de revertir una acción.

Cualquier software debería proveer de acciones de deshacer/rehacer para proporcionar al usuario la posibilidad de recuperarse de errores (por ejemplo, que el usuario elija una opción del sistema por equivocación). Para brindar esta característica esencial de usabilidad en procesos complejos, lentos o que produzcan grandes cambios en los estados de distintos

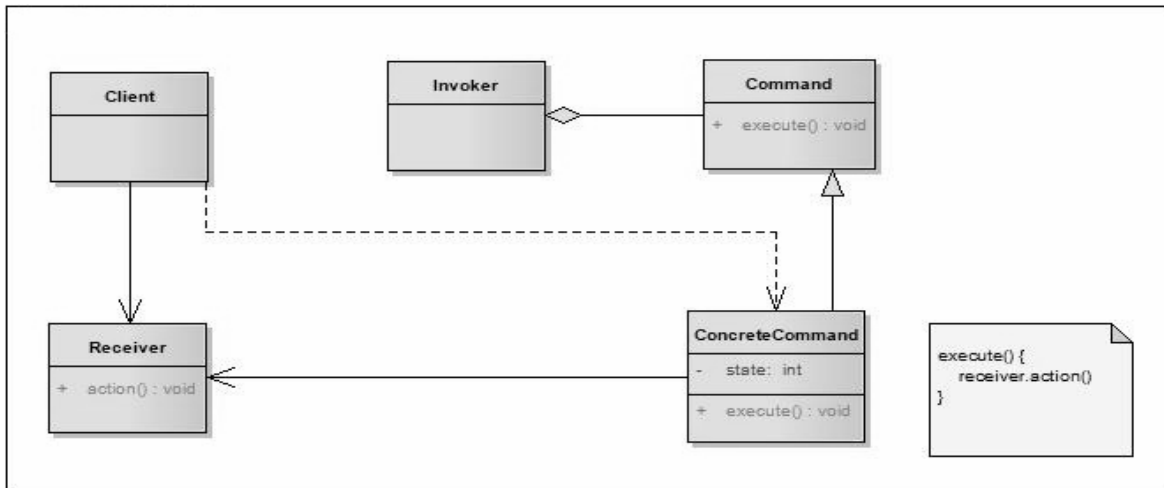


Figura 1. Patrón de diseño Command.

objetos, se modifican algunas clases del patrón Command, para implementar el patrón de usabilidad Undo [2].

A continuación, en la figura 1 se muestra un diagrama de clases para el patrón Command.

El patrón de usabilidad Undo fue implementado adaptando el patrón Command. Precisamente, la clase invocador, ahora llamada UndoRedo, será la encargada de contener la clase comando, que define los métodos execute() and unDo(). Las clases comando concreto, que existirá una por cada opción que ofrezca el software al usuario, heredan de la clase comando. A través del método execute() implementan comportamientos específicos que desencadenan una secuencia de acciones en el objeto receptor provocando cambios en este.

El método unDo() implementa los algoritmos inversos al método execute(). La clase Receptor representa la abstracción de varias clases, de las que surgirán objetos que participen en procesos críticos,

complejos o que demanden una gran cantidad de tiempo.

En efecto, el usuario requerirá, a medida que trabaje con el software, de rehacer/deshacer acciones. Por lo tanto, de acuerdo a la opción elegida por el usuario (clase Cliente), el software instancia comandos concretos para resolver dicha petición.

La clase UndoRedo, cuenta con una pila (stackCommands), en la cual acumula los comandos ejecutados a lo largo del tiempo. De tal manera, puede seguir la trazabilidad de los mismos. Si el software recibiera una o más peticiones de deshacer, el objeto UndoRedo tendrá disponible en el stack todos los comandos ejecutados. Por lo tanto, emplea los métodos Undo() para restablecer el o los objetos receptores a sus estados anteriores a la ejecución del método execute() de cada comando. En consecuencia, el usuario es capaz de navegar entre los diferentes estados del sistema, de acuerdo a las opciones que haya elegido.

El uso del patrón Command permitiría además implementar una opción para

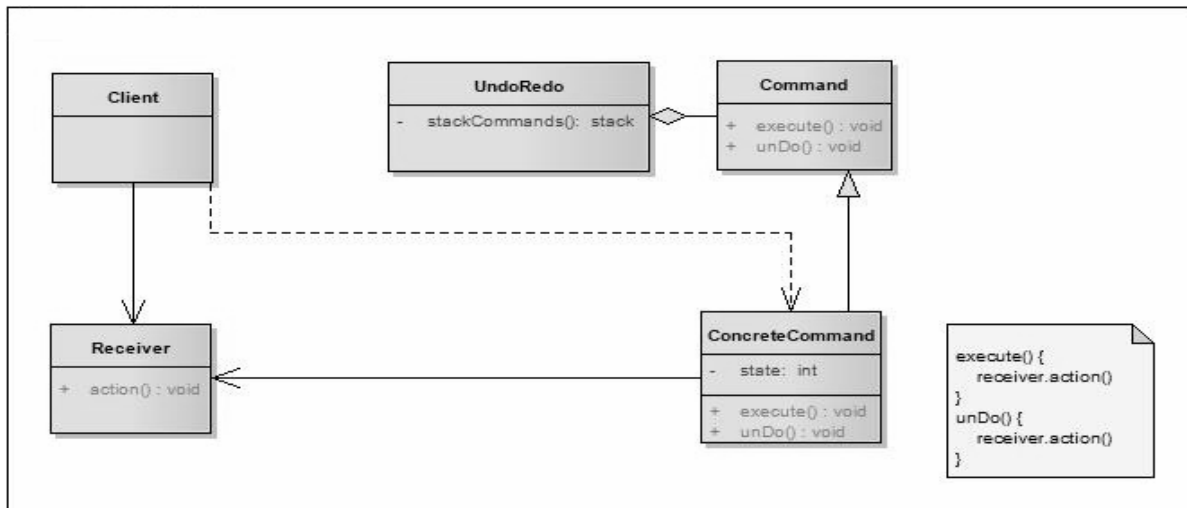


Figura 2. Patrón de usabilidad Undo incluido en el patrón de diseño Command.

cancelar procesos dentro del sistema, por ejemplo, procesos que demoren más tiempo del esperado. Esto se resuelve implementando en hilos separados los objetos UndoRedo y Receptor. Esto posibilita la ejecución asíncrona del algoritmo del receptor (finalizar el proceso que se encuentra en ejecución) y considerando lo explicado anteriormente, llevar el sistema al estado previo al proceso que se estaba ejecutando.

En síntesis, utilizando el patrón Command es posible implementar un componente dentro del sistema que se encargará de resolver los aspectos de usabilidad mencionados, de manera tal que no se deba lidiar con ellos en el futuro. Además, permite modularizar estados de procesos que realiza el sistema en los diferentes objetos comando concreto, lo que permite extender escalar el sistema de ser necesario.

Trabajos Relacionados

- **Patrones arquitectónicos sobre usabilidad en el dominio de las aplicaciones web [1]:** La principal contribución de este artículo es la

identificación de los patrones arquitectónicos y en qué medida éstos impactan en las propiedades de usabilidad en el contexto de las aplicaciones web. Este estudio demuestra que los patrones arquitectónicos tienen una influencia directa en la usabilidad propiamente dicha; y que la usabilidad de la aplicación web se ha mejorado al usarlos.

- **Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el momento de Arquitectónico [1]:** presenta una idea para mejorar la usabilidad de un sistema de software aplicando un proceso específico de diseño para usabilidad. Esta aproximación difiere de la idea tradicional de medir y mejorar la usabilidad una vez finalizado el sistema. Las soluciones de diseño que se obtuvieron por medio de este proceso garantizan que éstas son soluciones posibles, aunque no tienen por qué ser únicas.

Conclusión y Trabajos Futuros

Se ha visto que la aplicación del patrón fue lo que se esperaba ya que agilizo el proceso e introdujo la usabilidad de forma temprana dentro del proceso de software, lo cual abre la puerta para seguir estudiando la adaptación de patrones para construir los sistemas de información.

Los patrones de diseño y arquitectura pueden ser relacionados y modificados, para incluir aspectos de usabilidad desde la fase de diseño de software. En consecuencia, incluir aspectos de usabilidad no debería posponerse hasta la etapa de implementación del software. Se pueden reducir costos y tiempo de desarrollo, como así también la complejidad del sistema, aplicando soluciones de usabilidad, en una etapa de desarrollo temprana. Por otra parte, contemplar la usabilidad del sistema como un eje en su desarrollo en todas las etapas aumenta su calidad y, por lo tanto, la posibilidad de éxito del mismo.

Como trabajo futuro se propone investigar con mayor detalle el impacto que pueda tener la modificación de los patrones de Diseño, al incluir patrones de usabilidad y verificar el impacto de los cambios mediante la trazabilidad de los mismos.

Agradecimientos

Los autores del presente trabajo quieren manifestar su agradecimiento a la Secretaría de Ciencia, Tecnología y Posgrado de la Universidad Tecnológica Nacional, y al Ministerio de Ciencia y Tecnología de la provincia de Córdoba, mediante el Proyecto Apoyo a Grupos de Reciente Formación con Tutores (GRFT 2018 - Res. Nro: 000079 / 2018) del Programa de Generación de Conocimientos (PGC) 2018 por sus contribuciones para el desarrollo del presente trabajo de investigación.

Referencias

[1] Luis Arciniegas Herrera, María Verónica Fernández de Valdenebro, María Amparo Hormiga Juspian, Aleyda Tulande Arroyo, Cesar Alberto Collazos Ordoñez. Patrones arquitectónicos sobre usabilidad en el dominio de las aplicaciones web, INGENIERÍA E INVESTIGACIÓN VOL. 30 No. 1, ABRIL DE 2010.

[2] Ana M. Moreno & M. Isabel Sánchez, Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el momento de Arquitectónico.

[3] Elena, Ma & Hdz, Hernández & Alvarez Carrión, Guillermo & Muñoz-Arteaga, Jaime. (2003). Patrones de Interacción para el Diseño de Interfaces WEB usables.

[4] Paloma Díaz, Susana Montero, Ignacio Aedo. Ingeniería de la web y patrones de diseño.

Datos de Contacto

Ezequiel Enrique Fernandez.
CIDS, UTN, Facultad Regional Córdoba.
Maestro Marcelo López esq, Cruz Roja Argentina.
<ezequiel004@gmail.com>

Santiago Jose Solis Luna
CIDS, UTN, Facultad Regional Córdoba.
Maestro Marcelo López esq, Cruz Roja Argentina.
<sant.solis.luna@gmail.com>

Juan Pablo Fernandez Taurant
CIDS, UTN, Facultad Regional Córdoba.
Maestro Marcelo López esq, Cruz Roja Argentina.
<jtaurant@gmail.com>

Nicolas Perez
CIDS, UTN, Facultad Regional Córdoba.
Maestro Marcelo López esq, Cruz Roja Argentina.
<nicoperez444@gmail.com>

Juan Carlos Moreno
CIDS, UTN, Facultad Regional Córdoba.
Maestro Marcelo López esq, Cruz Roja Argentina.
<jmoreno33@gmail.com>

Marcelo Martin Marciszack
CIDS, UTN, Facultad Regional Córdoba.
Maestro Marcelo López esq, Cruz Roja Argentina.
<marciszack@gmail.com>