

IMPORTANCIA, PROBLEMAS Y SOLUCIONES EN EL DISEÑO DE LA INTERFAZ DE USUARIO

IMPORTANCE OF INTERFACE DESIGN - PROBLEMS AND SOLUTIONS

ALECIA ACOSTA; NANCY ZAMBRANO

*Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación,
Centro de Ingeniería de Software y Sistemas (ISYS.)
eacosta@strix.ciens.ucv.ve; nzambran@strix.ciens.ucv.ve*

RESUMEN

La Interacción Humano-Computador es una disciplina que abarca el diseño, implementación y evaluación del componente interactivo de las aplicaciones computacionales dirigidas a los humanos. Por su esencia, es un área multidisciplinaria donde convergen diversos tipos de especialistas. Este trabajo destaca la importancia de esta disciplina considerando el nuevo rol social de la informática en el contexto de los últimos años. El objetivo de este trabajo es esbozar algunos de los problemas que ocurren en el diseño de interfaces de usuario, los cuales surgen del carácter multidisciplinario de los equipos de desarrollo y las dificultades en su integración con la Ingeniería de Software, en la búsqueda de interfaces usables; así como el planteamiento de soluciones a los problemas anteriores a través del uso de los patrones de interacción en un método para la construcción de un prototipo de la interfaz de usuario. Este método integra el desarrollo de software y de la interfaz de usuario, y ha sido validado a través de casos de estudio que han permitido depurarlo y evidenciar su aplicabilidad y efectividad.

PALABRAS CLAVE: Interacción Humano-Computador, interfaz de usuario, método de desarrollo de interfaces, patrones de interacción, usabilidad, método de desarrollo orientado a objetos

ABSTRACT

Man-computer interactions underscore a discipline comprising the design, implementation, and evaluation of computer applications aimed at solving human endeavors. It is essentially a multidisciplinary area comprising a wide range of specialists. This paper highlights the importance of this discipline by dwelling on the social role of information technology within the context of technological breakthroughs in the last few years. The objective of this paper is to outline some of the problems occurring in user-interface design, which stem from the multidisciplinary character of the research and development teams and the difficulties in integration with software engineering in the search of usable interfaces, and to propose solutions to the aforementioned problems through interaction patterns aimed at the construction of user-interface prototyping, a method that integrates software development and user-interface, and which has been validated by case studies that have allowed for its applicability and effectiveness.

KEY WORDS: Man-computer interaction, user-interface, interface development method, interaction patterns, usability, object-oriented development method

INTRODUCCIÓN

La Interacción Humano-Computador (IHC) es una nueva disciplina que concierne con el componente interactivo entre las aplicaciones computacionales y los usuarios, así como las consideraciones del contexto de esta interacción. Es un área multidisciplinaria donde confluyen especialistas del dominio de la aplicación que se construye, especialistas en interfaces de usuario, psicólogos cognitivos, diseñadores gráficos, educadores y el usuario, entre otros (Borchers 2000).

La IHC es un área en permanente evolución dados los cambios tecnológicos que se suceden continuamente. Las expectativas de desarrollo en los

próximos años son tales que los grandes avances que se darán en la computación se centrarán en resolver los problemas de la interacción entre el humano y el computador.

En la actualidad han surgido nuevos roles de la informática; en particular el rol de la comunicación, el cual se sustenta en la revolución de las tecnologías informáticas y de las comunicaciones (TIC). Lo trascendental de las redes es que tras el hecho importante de la comunicación entre las máquinas está la comunicación entre las personas. Esta comunicación mediada por la tecnología, introduce como medio central la *interfaz* entre el humano y el computador. Justamente por ello, este nuevo rol trae como consecuencia la

importancia cada vez más creciente de las interfaces y, de manera general, de la Interacción Humano-Computador.

La *interfaz de usuario* es la parte accesible de un sistema computacional, tanto hardware como software, que permite al usuario interactuar con el sistema. El diseño de una buena interfaz se ha vuelto parte integral y relevante para las aplicaciones interactivas. Un sistema de software puede no tener aceptación si su interfaz es pobre (independientemente de que el software sea confiable y eficiente). El costo de desarrollo de una interfaz representa, en muchas aplicaciones, un porcentaje alto del costo de desarrollo de la aplicación.

La búsqueda constante de nuevas interfaces de usuarios está asociada a los avances tecnológicos, al rol social de la computación y se debe a que cada día se incrementa el número de usuarios con diferente formación, nivel social y habilidades; cada vez hay más información cuya complejidad va en crecimiento, cada vez es necesario ofrecer más servicios de diferente índole y cada vez se dispone de nuevas y poderosas tecnologías, las cuales avanzan de una manera vertiginosa. Todo ello establece un gran desafío para la Interacción Humano-Computador, la cual debe mantenerse a la par de este avance y sacar el mejor provecho de la tecnología para ponerlo a disposición de los usuarios.

El objetivo de este trabajo es presentar algunos de los problemas que surgen en el diseño de interfaces de usuario y las soluciones propuestas. Estos problemas se derivan del carácter multidisciplinario de los equipos de desarrollo y la falta de integración con la Ingeniería de Software, en la búsqueda de interfaces usables. En consecuencia, se plantea una solución basada en el uso de los patrones de interacción dentro de un método para la construcción de un prototipo de la interfaz de usuario.

La Sección 2 plantea algunos problemas relacionados con el diseño de las interfaces de usuarios. En la Sección 3 se define el concepto de patrones de interacción y se describe el método de construcción de interfaces de usuario como una solución a los problemas anteriores. Algunos de los resultados relevantes se presentan en la Sección 4 enfatizando la integración del método a un proceso de desarrollo de software. Luego se discuten estos resultados en la Sección 5, para finalizar con las principales conclusiones de esta investigación.

PROBLEMAS ACTUALES EN EL DESARROLLO DE INTERFACES DE USUARIO

El surgimiento del nuevo rol de la informática en la sociedad y la masificación del computador han traído como consecuencia el surgimiento de nuevos problemas en el diseño de interfaces de usuario, que se deben resolver. Entre éstos se distinguen tres problemas relevantes, asociados a:

a) La construcción de software usable

La usabilidad es una cualidad del software que está estrechamente ligada a la interfaz de la aplicación. De manera intuitiva, la usabilidad de un software está determinada por el grado de satisfacción del usuario cuando utiliza el producto. Una de las principales metas en la construcción de interfaces de usuario es la producción de sistemas *usables*. Nielsen (1993) establece que la interfaz de una aplicación es usable si posee los siguientes atributos: fácil de aprender, eficiente en cuanto al uso, fácil de memorizar, baja tasa de errores (que minimice los errores que pueda cometer el usuario) y que logre la satisfacción del usuario.

La usabilidad implica que debe tomarse en cuenta, en el diseño de la interfaz, a las personas que van a utilizar el sistema, lo cual puede ser una tarea compleja. De allí que sea crucial para lograr la usabilidad del producto, involucrar al usuario desde las primeras etapas del proceso de construcción de la interfaz de la aplicación. Para concretar, la usabilidad es la fuente de mayores iteraciones en el diseño de la interfaz, lo cual se traduce en gasto de tiempo y esfuerzo por parte del equipo de desarrollo.

b) El carácter multidisciplinario de los equipos de desarrollo

La Interacción Humano-Computador es un área inherentemente de naturaleza multidisciplinaria, el desarrollo de interfaces de usuario requiere de un equipo de trabajo que incluya especialistas en IHC y en Ingeniería de Software, especialistas en el dominio de la aplicación, y quizás, especialistas en otras áreas; incluyendo, además, la participación activa del usuario final. La producción de software usable va a depender, de que los especialistas del dominio y los usuarios finales participen en el diseño del producto. Es fácil imaginar los problemas de comunicación que pueden darse entre los miembros de un equipo como el anterior

debido a que, en general, ellos pueden hablar diferentes “idiomas” dependiendo de su profesión y experticia. El problema radica en que muchas veces la comunicación no fluye entre las partes, donde cada una tiene su propia experiencia y conocimientos.

c) La integración entre la Ingeniería de Software y la Interacción Humano-Computador

Una de las formas en que se manifiesta la falta de integración entre la Ingeniería de Software y la Interacción Humano-Computador es la ausencia del tratamiento de la construcción de la interfaz de usuario en los métodos de desarrollo de software más conocidos y usados; en particular, en el caso de los métodos orientados a objeto. En efecto, la mayoría de estos métodos reconocen la importancia del desarrollo de la interfaz de usuario pero adolecen de un tratamiento que integre el desarrollo de la interfaz y el desarrollo de la aplicación; hay una separación entre estos dos aspectos indisolublemente asociados de una aplicación; quizás en el pasado este hecho no era relevante; en la actualidad debido a la importancia que han adquirido

las interfaces de usuario, se pone de manifiesto la necesidad de contar con métodos que integren el desarrollo de la interfaz y de la aplicación.

En la sección que sigue se presentan algunas ideas para conectar estas dos disciplinas y un método para la construcción de interfaces que busca esa integración; en este método juega un papel preponderante el uso de patrones de interacción (Acosta 2004).

MÉTODOS Y HERRAMIENTAS LOS PATRONES DE INTERACCIÓN

Un *patrón de interacción* (o patrón de interfaz) describe una solución exitosa a un problema recurrente concerniente a la interfaz de usuario, en un contexto dado. Un patrón de interacción es un medio de comunicación que expresa una solución de interfaz en una forma textual y gráfica, a fin de ser entendida por otras personas, con diferentes formaciones profesionales. La Figura 1 muestra un ejemplo de un patrón de interacción, tomado de la colección de patrones de Van Welie (2005), disponible en el Web.

Nombre	Formatos de datos de fechas Martijn van Welie
Problema	El usuario desea introducir datos de fechas y no desea preocuparse por la sintaxis del dato
Solución	Permitir que el usuario elija la fecha de un calendario tal como se encuentra en el mundo real y que sólo realice selección -el usuario no tipea.
Contexto	Todos los sistemas que requieran que el usuario introduzca fechas (importante en interfaces internacionales)
Fuerzas	- Los datos de fechas tienen múltiples sintaxis - Convenciones culturales determinan la sintaxis esperada
Usabilidad	Guiar al usuario y prevenir errores
Ejemplo	Panel en Windows para la búsqueda de un fichero creado a partir de una fecha que introduce el usuario (por selección) 

Figura 1: Un ejemplo de un patrón de Interacción denominado *Formato de datos de fechas*

Un patrón captura la experiencia y conocimiento de expertos, quienes han producido soluciones exitosas a problemas, a fin que esas soluciones queden a disposición de personas con menos experiencia; sin embargo, los patrones no proveen siempre las soluciones definitivas, algunas veces los usuarios de patrones deben adaptar, utilizar, instanciar o implementar un patrón.

En la literatura se encuentran diversas formas de escribir un patrón de interacción Erickson (2005 a), las cuales coinciden en algunos componentes básicos. En la actualidad no existe una estructura estándar para describir un patrón de interacción. En Borchers (2000) se propone una estructura de patrón haciendo uso de un metapatrón, esto es, un patrón que permite describir patrones. En este metapatrón se definen los componentes requeridos para describir los patrones, en cualquier dominio de conocimiento. Esta estructura se presenta en la Tabla 1. No siempre es necesario describir cada uno de los componentes para todos los patrones, salvo aquellos componentes obligatorios: nombre, problema, solución y contexto.

Tabla 1: Estructura de los patrones de interacción

Componente	Descripción
Nombre	Comunica la idea central del patrón.
Autor	Indica el nombre del creador del patrón.
Clasificación	Indica el tipo de patrón.
Rango	Indica la calificación del patrón.
Problema	Describe la interacción desde el punto de vista del usuario
Solución	Describe, en forma descriptiva y gráfica, la solución del problema
Contexto	Presenta las condiciones bajo las cuales se usa este patrón
Fuerzas	Señala los conflictos que pueden restringir la solución
Usabilidad	Describe el impacto del uso del patrón para el usuario.
Consecuencias	Describe el resultado de aplicar el patrón.
Ejemplos	Muestra ejemplos de la solución propuesta.
Patrones relacionados	Enumera otros patrones relacionados con este patrón (cuando un patrón referencia a otro significa que requiere de su descripción para completar la solución)

En términos del diseño de la interfaz de una aplicación, lo que interesa es la colección de patrones que describen todos los problemas de interacción.

Los patrones de interacción permiten: describir un problema de interacción, su contexto y la solución; así como cierta información adicional; expresar una solución que ha demostrado ser exitosa; promover el reuso de soluciones; facilitar la comunicación entre miembros de equipos pertenecientes a diferentes disciplinas; registrar el conocimiento y la experiencia, que queda a disposición de otros; generalizar una solución.

Los patrones de interacción se presentan como un concepto que puede ayudar a encontrar soluciones a los problemas planteados al inicio de este artículo, debido a que:

Los patrones deben concebirse teniendo en mente la usabilidad; esto es, un patrón de interacción se centra en soluciones que mejoran la usabilidad, mostrando el impacto en los indicadores correspondientes.

Los patrones pueden facilitar la comunicación necesaria entre los miembros del grupo de desarrollo de la interfaz. Diversos autores han trabajado en la creación de una “lingua franca” Erickson (2005 a) que facilite esta comunicación, basada en patrones. Ello significa que a un cierto dominio de interés (por ejemplo, comercio electrónico, cursos virtuales, aplicaciones *groupware*, etc.), se le asocia una colección de patrones, que describen las interacciones a diferentes niveles de abstracción del dominio. De esta manera ellos conforman un lenguaje de patrones que en cierta forma constituye una ontología del dominio y que concierne con la reutilización del conocimiento en ese dominio. De allí que entre las características de los patrones es importante que sean comprensibles y, además, estén organizados de una manera tal que se facilite su búsqueda y aplicación.

Los patrones de interacción pueden ser utilizados para modelar los elementos de la interfaz de usuario, lo cual puede formar parte de un proceso de desarrollo de software que incorpore la construcción de la interfaz desde las primeras etapas del ciclo de vida de una aplicación interactiva. Esta modelación permitiría describir claramente los elementos que deben formar parte de la interfaz y, luego, se implementan las soluciones propuestas en los patrones, para construir el prototipo de la interfaz de usuario.

En la actualidad, en Internet se dispone de repositorios o bases de datos de patrones de interacción -Erickson (2005 b); Van Welie (2005)-, en la forma de colecciones,

catálogos o lenguajes, que recogen la experticia de los diseñadores o el conocimiento de los expertos. Así, en la práctica, para el desarrollo de una aplicación en un determinado dominio es posible contar con la experiencia recopilada en estos patrones, examinar y analizarlos a fin de reusarlos. Por ejemplo, en el dominio Comercio Electrónico, se dispone de la colección de patrones desarrollados por Van Welie *op. cit.*

A continuación se describe un método para la construcción de la interfaz de usuario, basado en patrones de interacción. El método para la Construcción de la Interfaz de Usuario que se propone puede integrarse en un proceso de desarrollo de software orientado a objetos. Este método se inscribe en los modelos de desarrollo que incorporan el prototipaje desde las primeras etapas del ciclo de vida de los sistemas.

El objetivo principal del método consiste en el desarrollo de un prototipo de la interfaz de usuario, generado a partir de los patrones de interacción. Este prototipo puede ser evaluado por otros expertos del grupo multidisciplinario y por el usuario y, así, el prototipo puede evolucionar hasta satisfacer los requerimientos del usuario, definiendo un proceso de ingeniería de requerimientos basado en patrones.

Para la construcción del prototipo, previamente se desarrollan los siguientes modelos definidos en UML (*Unified Modeling Language*) (Rumbaugh *et al.* 2005):

El modelo de Casos de Uso el cual define todos los escenarios posibles a partir de las interacciones del usuario y muestra el comportamiento del sistema.

El modelo Objeto del Dominio que permite identificar los objetos que forman parte del dominio del sistema y las relaciones entre éstos.

Se hace énfasis en que este método aborda únicamente los aspectos referentes a la construcción de la interfaz, en consecuencia, debe ser incorporado a un proceso de desarrollo de software orientado a objetos que considere el resto de las etapas del ciclo de vida de una aplicación.

Hacia la construcción de la Interfaz de usuario

La Figura 2 muestra los modelos requeridos (Casos de Uso y Objeto del Dominio) y los modelos que se crean durante el proceso de construcción de la interfaz: un modelo Objeto de Interfaz y un Lenguaje de Patrones, a partir de los cuales se desarrolla un Prototipo.

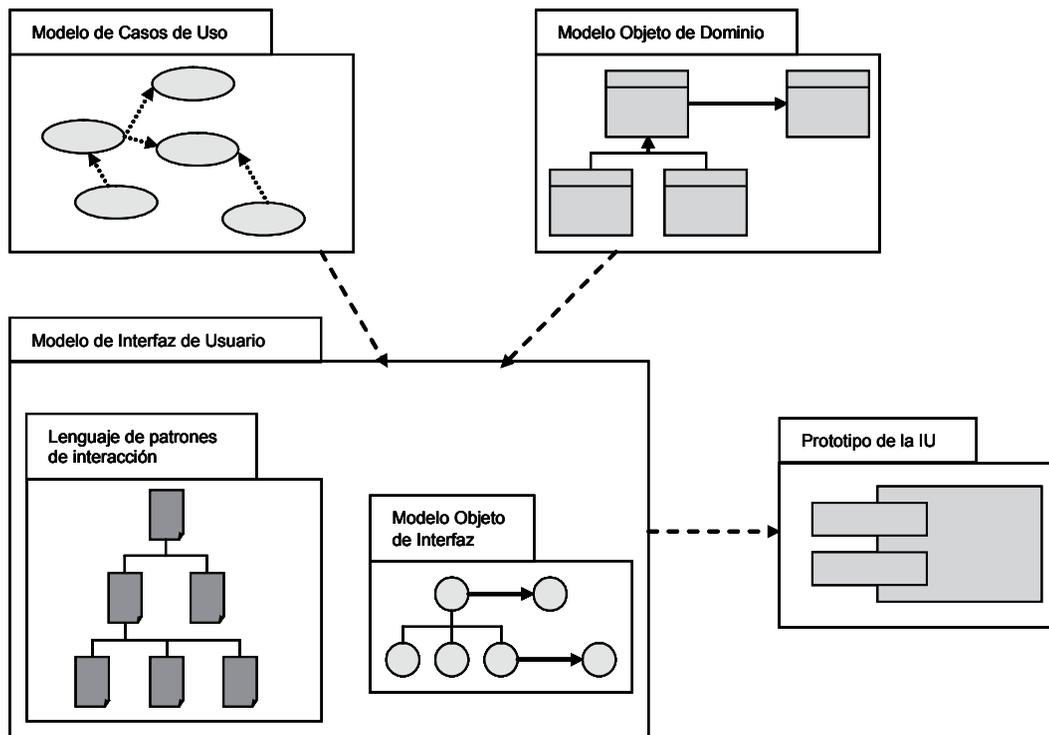


Figura 2: . Modelos que intervienen en la construcción del prototipo de interfaz de usuario

El Lenguaje de Patrones de Interacción Van Welie y Van Der Veer (2005) está formado por un conjunto de patrones que describen las diversas interacciones del usuario con el sistema, y que conforman un grafo. Así, el patrón del sistema (que describe el propósito del sistema, y los aspectos de usabilidad que se deben tomar en cuenta en el diseño de la interfaz) referencia a los patrones de tareas (que describen las interacciones de las funcionalidades) y éstos referencian a patrones de componentes de interfaz. Para completar el lenguaje, se crea un patrón de dominio que describe el dominio al cual pertenece el sistema que se está desarrollando.

El modelo Objeto de Interfaz está conformado por los objetos presentes en la interfaz, algunos provienen del modelo del dominio (aquellos que requieren una representación en la interfaz); otros son nuevos objetos que resultan de necesidades propias de la interfaz para resolver las interacciones (por ejemplo, la operación cortar

en un editor se representa, generalmente, mediante un objeto tijera). En ambos casos el resultado es producto de una reificación, que consiste en el proceso de conversión de los objetos del modelo conceptual del dominio a sus representaciones concretas en el modelo objeto de interfaz (Beaudouin-Lafon y Mackay 2000).

La Figura 3 muestra un ejemplo de reificación partiendo del modelo del dominio, en el cual, se ha definido la clase Documento (en un cierto dominio de manipulación de documentos, carpetas, etc.); En esta figura, a la izquierda se tiene la representación en la interfaz del objeto *documento*. A la derecha, se tiene la representación en la interfaz de una operación que se realiza sobre un documento: *abrir*. Note que en el primer caso se trata de un objeto simple de interfaz (documento) y en el otro caso corresponde a una ventana (abrir), ambos descritos mediante patrones que conforman el lenguaje de patrones que describe esta interfaz.

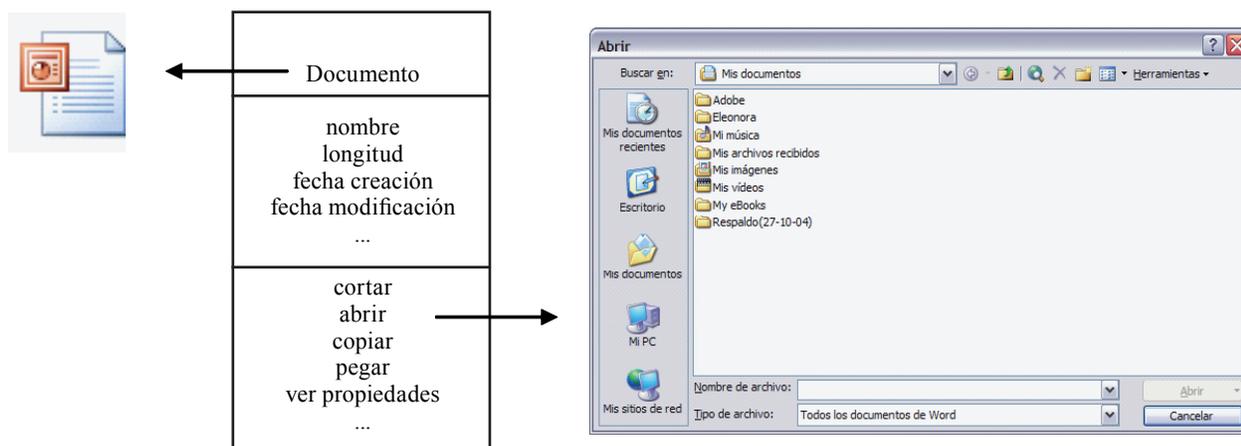


Figura 3: Un ejemplo de reificación del objeto *Documento* y de la operación *Abrir*

Proceso para la construcción del prototipo de interfaz

A continuación se describen las actividades que se realizan en este proceso.

a) El primer paso consiste en la elaboración del Lenguaje de Patrones de Interacción a partir del análisis del modelo de casos de uso y/o del modelo objeto del dominio, ello permite detectar los puntos de interacción del usuario con la aplicación; cada uno de estos puntos

(o varios) se traduce en un patrón que le asocia el problema de interacción a resolver, desde el punto de vista del usuario. Estos patrones pueden completarse referenciando a otros patrones de interacción más simples. Al final, se tendrá un conjunto de patrones relacionados que conforma un lenguaje de patrones, el cual describe la interfaz de la aplicación, constituyendo por tanto un prototipo de la interfaz. Los patrones de interacción constituyen el concepto fundamental y novedoso para la definición del método propuesto; esta parte del proceso consiste en describir cada uno de los componentes del

patrón (Nombre, Problema, Contexto, Solución -textual y gráfica-, Consecuencia, Patrones relacionados, etc.) buscando las soluciones apropiadas.

b) El segundo paso corresponde a la elaboración del Modelo Objeto de Interfaz, para ello se toma como base el modelo Objeto del Dominio y se aplica la *reificación* de los objetos del dominio que se requieran en la interfaz y se crean nuevos objetos mediante la reificación de operaciones, para cada una de las cuales se define un *instrumento de interacción*, esto es, un mediador entre el usuario y el objeto en la interfaz (Beaudouin-Lafon y Mackay 2000).

c) La construcción del prototipo de interfaz de usuario parte del análisis del Lenguaje de Patrones y del modelo Objeto de Interfaz. Se pueden presentar varios casos dependiendo de la forma en que se exprese la solución en los patrones de interacción, estos casos son:

- Si las soluciones contemplan un bosquejo de las pantallas, o se expresan en forma descriptiva, por ejemplo diagramas hechos a mano sobre papel, se tendría un prototipo de interfaz de baja fidelidad, a partir del cual, se debe construir un prototipo de alta fidelidad implementado las soluciones descritas en los patrones.

- Si las soluciones de los patrones se expresan gráficamente, a través de un diseño de interfaz que ha sido realizado utilizando el lenguaje de implementación del producto final, se tiene prácticamente elaborado un prototipo de alta fidelidad.

RESULTADOS APLICACIÓN DEL MÉTODO

El método para la construcción de la interfaz constituye un resultado que permite resolver los problemas planteados en la Sección 2, ya que los modelos que se construyen en la aplicación del método (el lenguaje de patrones de interacción y el modelo objeto de interfaz) constituyen la base para la construcción del *prototipo de la Interfaz de Usuario*. Se trata de un prototipo de alta fidelidad, el cual se genera mediante un prototipaje horizontal y evolutivo.

Este método inserta el prototipaje de la interfaz en el proceso de desarrollo de software (desde su fase inicial) y puede ser integrado a cualquier proceso de desarrollo de software orientado a objetos, que se soporte en tres principios básicos:

- dirigido por casos de uso: los casos de uso dirigen y controlan el proceso en su totalidad,

- centrado en arquitecturas,

- que sea iterativo e incremental: el desarrollo se plantea de manera progresiva, por incrementos regresando a etapas anteriores, hasta lograr el producto final.

Como notación se utiliza el Lenguaje de Modelación Unificado, UML Rumbaugh *et al.* (2005), para expresar los diagramas que describen distintas perspectivas de una aplicación.

Se debe hacer énfasis que, en este método, se propone la construcción de un prototipo de alta fidelidad, de tal forma que éste evolucione hasta el prototipo ejecutable mediante un proceso evolutivo de desarrollo. En efecto, una vez que se tiene el prototipo de alta fidelidad, lo que falta es la integración y la implementación de la navegación entre las pantallas y el resto de los detalles a fin que este artefacto sea ejecutable y pueda continuar evolucionando hasta convertirse en el producto final. Una vez construido el prototipo ejecutable, se continúa el desarrollo de la aplicación siguiendo cualquier proceso orientado a objeto.

Por último, se señala la importancia del uso de componentes reusables en el proceso de construcción en las diferentes fases del ciclo de vida del software, tanto patrones de interacción para el prototipaje de la interfaz, como patrones de análisis o patrones de software. Según algunos autores Carroll (2002) el diseño mediante el reuso se convertirá en el paradigma de desarrollo de software dominante. Sin embargo hay que agregar que la manipulación eficiente de componentes requiere el uso de herramientas que permitan la creación, la recuperación, la adaptación, etc. de estos componentes; ello en la búsqueda de procesos de construcción de software mediante ensamblaje de componentes.

DISCUSIÓN

Actualmente se encuentra en la literatura investigaciones orientadas a la integración de la construcción de la interfaz de usuario al proceso de desarrollo de software, por ejemplo algunos de los trabajos presentados en la Conferencia Latinoamericana de Interacción Humano-Computador CLIHC 2003 y 2005 (Greco *et al.*, 2005).

Durante el desarrollo de esta investigación se experimentó el método con varios casos de estudio, mostrando su simplicidad y efectividad para la producción de un prototipo de la interfaz de usuario; así como también su incorporación en métodos de desarrollo de software. Uno de estos casos de estudio consistió en el desarrollo de un Sistema de Modelado de Medios Geológicos, denominado *SistGeo* Reyes *et al.* (2002), el cual permite solventar un problema fundamental en la Geofísica que consiste en la modelación de las características de una región específica del subsuelo y su posterior manipulación. Otra experiencia interesante se experimentó con el curso de Ingeniería de Software del 3^{er} semestre de la Licenciatura en Computación, al que se le explicó los conceptos y terminología básica necesaria para aplicar el método; posteriormente se le asignó la tarea de diseñar una interfaz de usuario de un sistema interactivo, proporcionándole los modelos de Casos de Usos y Objetos del Dominio correspondientes. Así mismo, esta técnica fue aplicada por otro grupo de estudiantes del 5^o semestre en la asignatura Interacción Humano-Computador a quienes se les dictaron clases relacionadas con el tema y se les asignó una tarea similar. Estas experiencias permitieron mostrar la simplicidad del método y permitió enriquecerlo.

CONCLUSIONES

En este artículo se destaca la importancia de las interfaces y la necesidad de métodos para el diseño de interfaces de usuario que se apliquen desde el inicio del proceso de desarrollo de software y que se integren al mismo, así como la participación de diferentes especialistas en el desarrollo de la interfaz de usuario. Se propone un método que debe conjugar los intereses de los diferentes actores en el diseño: a los desarrolladores, a los especialistas en el dominio y al usuario. Un concepto que facilita esta premisa son los patrones de la interacción y constituyen la base para la definición del método propuesto para la construcción de la interfaz de usuario. De esta manera, el método permite la construcción de interfaces usables, facilitando la comunicación entre los miembros del equipo de desarrollo e integrando la Interacción Humano-Computador a la Ingeniería de Software; con lo cual se satisface el objetivo planteado al inicio de esta investigación.

En cuanto a los patrones de interacción, se propone una estructura que facilita su comprensión, tanto para los miembros del equipo multidisciplinario de desarrollo como del usuario, desde el inicio del proceso. Esta estructura define los términos que se utilizan para

representar un área de conocimiento mediante patrones, lo cual define un lenguaje de patrones.

Una de las ventajas principales de los patrones de interacción consiste, de hecho, en reusar las soluciones exitosas a problemas concernientes a la construcción de interfaz de usuario, esto es, reusar el conocimiento de un cierto dominio concerniente al desarrollo de interfaces.

El método propuesto integra en una unidad el desarrollo del software y el desarrollo de la interfaz, realizando el prototipaje de la interfaz, al inicio mismo del proceso, como parte de la fase de análisis de cualquier método orientado a objeto (Larman 2002). El método se ha validado de manera experimental con desarrolladores de diversa formación, con resultados satisfactorios, tanto en cursos de la Licenciatura de Computación en la UCV, como en tesis de pregrado y postgrado (Pedraza 2003). Estos casos de estudio permitieron simplificar y depurar el método, poniendo en evidencia la aplicabilidad y efectividad del mismo.

AGRADECIMIENTOS

Las autoras de este trabajo agradecen al Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela (CDCH-UCV), por el financiamiento otorgado para la realización de esta investigación.

REFERENCIAS BIBLIOGRÁFICAS

- ACOSTA A.E. 2004. Métodos y Herramientas para la construcción de la Interfaz de Usuario utilizando Patrones de Interacción (Tesis de Doctorado). Universidad Central de Venezuela, Facultad de Ciencias, p.186.
- BEAUDOUIN-LAFON M.; MACKAY W. 2000. Reification, Polymorphism and Reuse: three principles for designing visual interfaces. Proc. Advanced Visual Interfaces, Italy, pp. 102-109
- BORCHERS J. 2000. Breaking the interdisciplinary limits of computer-human interaction design: A pattern approach. SIGCHI. Vol 32 1.
- BORCHERS J. 2001. A Pattern Approach to Interaction Design. Jhon Wiley, p. 245
- CARROLL J. 2002. Human-Computer Interaction, the New Millenium. Addison-Wesley. USA

- ERICKSON T. 2005a. Lingua franca for design: sacred places and pattern languages". URL: <http://www.visi.com/~snowfall/DesignLinguaFranca.html> (Consultado Junio 2005)
- ERICKSON T. 2005b. Patterns Home Page. URL: <http://www.visi.com/~snowfall> (Consultada Junio 2005).
- GRECO M., JUNQUEIRA S.; LUCENA C. 2005. Conveying Human-Computer Interaction Concerns to Software Engineers Through an Interaction Model. Conferencia Latinoamericana de Interacción Humano-Computador, CLIHC 2005.
- LARMAN C. 2002. Applying UML and patterns: An introduction to object-oriented analysis and design and the unified process. Second Edition. Prentice Hall PTR, p. 627. USA
- NIELSEN J. 1993. Usability Engineering. Prentice-Hall. USA., pp. 23-37.
- PEDRAZA F. 2003. Genasig: un Generador de Sitios Web de Asignaturas (Trabajo Especial de Grado). Licenciatura de Computación. UCV.
- REYES, A.; ACOSTA, E.; ZAMBRANO, N. 2002. Patrones de Interacción: su uso en la construcción de la Interfaz de una Aplicación de Modelación de Medios Geológicos. RT 2002-01. Lecturas en Ciencias de la Computación, ISSN 1316-6239. Escuela de Computación, UCV, Caracas, 2002.
- RUMBAUGH J., BOOCH G.; JACOBSON I. 2005. Unified Modeling Language User Guide. Addison-Wesley Professional, 2nd edition.
- VAN WELIE M. 2005. Patterns Collection. URL: www.welie.com/patterns (Consultado Junio 2005)
- VAN WELIE M.; VAN DER VEER G. 2005 Pattern Languages in Interaction Design: Structure and Organization. En Interact'03. Zurich. Eds. Rauterberg, Menozzi, Wesson IOS Press, Amsterdam. p. 527-534. 2003. URL: <http://www.welie.com/patterns/literature.html> (Consultado junio 2005)