

PROTOTIPAJE DE LA INTERFAZ DE USUARIO: UNA EXPERIENCIA EN EL DOMINIO DE LA GEOLOGÍA

ALECIA ELEONORA ACOSTA¹, ALDO REYES², NANCY ZAMBRANO¹

¹ Universidad Central de Venezuela. Facultad de Ciencias. Escuela de Computación. Centro de Ingeniería de Software y Sistemas (ISYS) Caracas, Venezuela. e-mails: eacosta@ciens.ucv.ve nizambran@ciens.ucv.ve

² Universidad de Carabobo. Facultad Experimental de Ciencias y Tecnología.
Dpto. de Ciencias de la Computación, Valencia, Venezuela. e-mail: reyes@uc.edu.ve

Recibido: enero de 2007

Recibido en forma final revisado: febrero de 2008

RESUMEN

En este trabajo se describe la utilización de los patrones de interacción, incorporados a un método de desarrollo de software orientado a objetos, para la construcción de un prototipo de la interfaz de usuario perteneciente a una aplicación de Modelación de Medios Geológicos, lo cual significó una experiencia de desarrollo de una herramienta inscrita en el dominio de la geología. Los patrones de interacción, en conjunción con el modelo objeto de interfaz, facilitan la construcción de este prototipo de interfaz desde las primeras etapas del ciclo de vida de la aplicación. Un patrón de interacción captura un problema de interacción, una solución exitosa y su contexto. El patrón se escribe para comunicar una experiencia y permitir su reutilización. En este trabajo se muestra la factibilidad y efectividad de aplicar un método para la construcción de un prototipo de interfaz de usuario, incorporado al Proceso Unificado de Desarrollo de Software y utilizando el Lenguaje de Modelación Unificado (UML) para expresar los modelos que describen la aplicación.

Palabras clave: Interfaz de usuario, Patrones de interacción, Método de desarrollo de software orientado a objetos, Modelación de medios geológicos.

USER INTERFACE PROTOTYPING: AN APPLICATION IN GEOLOGY

ABSTRACT

In this work we describe interaction patterns and their incorporation into an object-oriented software development method in order to construct a user interface prototype. In this case, this prototype belongs to a geologic media modeling application. Interaction patterns, in conjunction with an interface object model, facilitate the construction of a user interface prototype, starting from the first stages of the software life cycle. An interaction pattern describes an interaction problem, a solution and its context. A pattern is written in order to communicate experience and knowledge, and reuse it. In this work, we show feasibility and effectiveness to apply a user interface construction method, incorporated to the Software Development Unified Process and using the Unified Modeling Language to express application models.

Keywords: User interface, Interaction patterns, Object-oriented software development method, Geologic media modeling.

INTRODUCCIÓN

En este artículo se describe la aplicación de un método para la construcción de la interfaz de usuario, el cual se integra al proceso de desarrollo de software. Se elige como caso de estudio un sistema de modelación de Medios Geológicos. Este método se basa en la construcción de un prototipo de interfaz de usuario una vez identificados los requerimientos del sistema y, para ello se elaboran dos modelos o visiones

diferentes del mismo sistema: un Lenguaje de Patrones de Interacción y el Modelo Objeto de Interfaz, los cuales van a facilitar la construcción de este prototipo. El método parte del Modelo de Casos de Uso y del Modelo Objeto del Dominio resultados del análisis de la aplicación (previsto en cualquier método de desarrollo de software orientado a objeto). Luego, el proceso de prototipaje de la interfaz se integra como una etapa en el desarrollo de software orientado a objetos. Este método es definido en Acosta

(2004) y el caso de estudio es descrito en Reyes (1999).

MODELACIÓN DE MEDIOS GEOLÓGICOS

Un problema fundamental en la Geofísica lo constituye la determinación de las características de una región específica del subsuelo. Tal determinación es muy importante en exploración petrolera, ya que conocer la forma de las superficies interfases que separan regiones con diferentes tipos de roca y determinar la densidad de estas regiones, proporciona información valiosa para identificar zonas del subsuelo donde pueden haberse acumulado hidrocarburos. Se desea, entonces, desarrollar un sistema que permita la construcción de modelos de medios geológicos y que sea interoperable con aplicaciones de trazado de rayos sísmicos. En lo sucesivo, este sistema será referido como *SistGeo*.

SistGeo debe permitir al usuario modelar diversas particularidades que pueden presentarse en medios reales y disponer de una interfaz gráfica de usuario que permita, en todo momento, la visualización y manipulación directa, en lo posible, del modelo tridimensional del medio en construcción (en lo sucesivo se utilizará el término «medio» para designar al modelo tridimensional del medio). Esta interfaz debe facilitar la interacción del usuario, quien podrá rotar, acercar o alejar el medio, modificando la visualización. También debe facilitar las restantes funcionalidades que debe proporcionar el sistema, identificadas durante el levantamiento de los requerimientos del mismo. Para satisfacer estos requerimientos, la interfaz debe acercarse lo más posible al modelo mental del usuario -sea el geofísico, el geólogo o el ingeniero que va a utilizar el sistema-, proporcionando e integrando al sistema los elementos de ayuda (por ejemplo, uso de tooltip, información contextual y ayuda general), evitando la memorización de comandos, simplificando las operaciones. En definitiva, se busca desarrollar un producto confiable, robusto y con un alto grado de usabilidad.

Los indicadores de usabilidad, tal como lo propone Nielsen (1993), se centran en que los sistemas sean fáciles de aprender, fáciles de memorizar, permitan al usuario ser productivo lo antes posible (esto es, eficiente en cuanto al uso), eviten que el usuario cometa errores y, en caso de ocurrir, indiquen las vías de recuperación de situaciones erróneas o de excepción; y que el usuario se sienta satisfecho cuando utilice la aplicación. Para ello, el diseño de la interfaz de usuario debe, necesariamente, estar centrado en el usuario a fin de garantizar que el sistema interactivo cumpla esta cualidad de usabilidad.

El análisis de requerimientos se realizó a través de entrevistas y discusiones con usuario potenciales de la aplicación, característica fundamental del diseño centrado en el usuario.

A continuación se muestra el resultado de este análisis de requerimientos mediante un listado de las funcionalidades de *SistGeo*; indicándose la funcionalidad y su descripción:

- Creación de Medio: crea el modelo de un medio formado por un número especificado de regiones uniformes, separadas entre sí mediante superficies interfases planas, dentro de un paralelepípedo rectangular de dimensiones dadas.
- Traslación de Superficie: traslada verticalmente una superficie interfase, modificando su profundidad.
- Inclinación de Superficie: inclina una superficie interfase, modificando la profundidad de uno de sus bordes.
- Traslación de Nodo: traslada un nodo, modificando sus coordenadas (x, y, z).
- Traslación de Línea de Nodos: traslada en forma paralela una línea de nodos, modificando en forma simultánea las coordenadas de todos los nodos de esa línea.
- Inserción de Línea de Nodos: inserta una línea de nodos en el polígono de control de una superficie, en una posición especificada.
- Duplicación de Superficie: construye una copia de una superficie y la coloca a una distancia específica de la superficie original.
- Fusión de Superficie: traslada una región de una superficie origen y la hace coincidir con otra región de una superficie destino, construyendo para esto, parches en la superficie origen que permitan hacer la conexión.
- Eliminación de Superficie: elimina una superficie o una región de una superficie, eliminando así la frontera entre dos o más regiones.
- Creación de Fractura: crea una fractura en una superficie, desplazando verticalmente parte de la misma.

CONSTRUCCIÓN DE LA INTERFAZ DE USUARIO

La idea central es la integración del proceso de construcción de la interfaz de usuario en un método de desarrollo de software orientado a objeto existente (basado en las funcionalidades del sistema, fundamentado por diferentes arquitecturas, iterativo e incremental). Esto se traduce en la incorporación de una fase de prototipaje de la interfaz de usuario en cualquier método orientado en objeto; en particular, en el Proceso Unificado de Desarrollo de Software

propuesto en Larman (2002). Adicionalmente se utiliza UML (Lenguaje de Modificación Unificado) como notación para la modelación del sistema (Rumbaugh *et al.*, 2005).

En el método propuesto, los modelos básicos que sirven de punto de partida para la construcción del prototipo son el Modelo de Casos de Uso y el Modelo Objeto del Dominio. Cada uno presenta una visión del sistema desde una perspectiva diferente y se expresa, para su visualización, mediante los diagramas correspondientes definidos en UML:

- El Modelo de Casos de Uso permite identificar los puntos de interacción entre el usuario y el sistema, con base en sus funcionalidades. Un Caso de Uso es una secuencia de transacciones iniciada por el usuario y expresa un comportamiento del sistema. El modelo completo define todos los escenarios posibles, a partir de las interacciones del usuario y muestra el comportamiento del sistema.

- El Modelo Objeto del Dominio permite identificar los objetos que se encuentran en el dominio del problema y las relaciones entre ellos. La elaboración de este modelo se realiza a partir del Modelo de Casos de Uso, lo cual permite integrar estas dos visiones del sistema.

Estos modelos son la base para elaborar, posteriormente, el Modelo de la Interfaz de Usuario (conformado por una colección de Patrones de Interacción y un Modelo Objeto de Interfaz) y éste es el punto de partida para la construcción del prototipo de interfaz. No se trata, pues, de proponer un método nuevo, ni se parte de cero; el proceso de construcción de la interfaz se integra a un método de desarrollo de software orientado a objetos.

Construcción del prototipo de la interfaz de usuario

El punto clave es incorporar la construcción de la interfaz de usuario al proceso de desarrollo del software, para ello se introduce el concepto de patrón de interacción que va a facilitar, conjuntamente con los objetos de la interfaz, el desarrollo del prototipo de la interfaz.

En general, un patrón de interacción captura la información esencial relativa a un problema de interacción frecuentemente encontrado, muestra una solución exitosa y describe el contexto de esta solución; éste se escribe para comunicar esa experiencia y permitir su reutilización. En particular, un patrón de interacción modela un aspecto referente a la interfaz de usuario. Los patrones sirven para generalizar una solución, para registrar el conocimiento y la experiencia y reusarlo, siendo útil para facilitar la comunicación entre las personas involucradas en el desarrollo de un sistema,

generalmente pertenecientes a diferentes disciplinas (el ingeniero de software, el especialista en interacción humano-computador, el experto en el dominio del problema, -geólogos en el ejemplo-, etc.). En la figura 1 se muestran los diferentes modelos que intervienen para la construcción del prototipo.

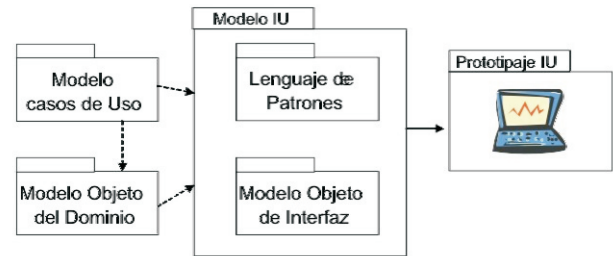


Figura 1. Modelos que intervienen en la construcción del prototipo de interfaz.

El primer modelo -Casos de Uso- permite detectar los puntos de interacción entre el usuario y la aplicación; cada uno de estos puntos (o varios) se traduce en un patrón que le asocia el mecanismo de interacción correspondiente, cuyo centro es 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 que definen un lenguaje en un cierto dominio, cuyas relaciones establecen una estructura enlazada.

El segundo modelo -Objeto del Dominio- es la base para la *reificación* o proceso de conversión de los conceptos presentes en este modelo, a sus representaciones concretas en la interfaz de usuario, generándose el Modelo Objeto de Interfaz. Tal como se define en Beaudouin-Lafon y Mackay (2000) la reificación consiste en el proceso de conversión de los conceptos en un determinado modelo a sus representaciones concretas en otro modelo de más bajo nivel de abstracción; tal reificación permite crear las representaciones a ser manipuladas por el usuario en la interfaz. Los objetos que pertenecen al Modelo Objeto de Interfaz provienen, entonces, de la reificación de objetos del Modelo del Dominio o de la reificación de operaciones de estos objetos, para las cuales se define un instrumento de interacción o mediador entre el usuario y un objeto de la interfaz. Como analogía considérese una operación común, por ejemplo la escritura, ello requiere de un instrumento para realizarla, por ejemplo, un lápiz sería el instrumento de interacción para realizar la escritura.

Las soluciones descritas en la Colección de Patrones y el Modelo Objeto de Interfaz, se implementan para obtener el prototipo de la interfaz de usuario.

APLICACIÓN DEL MÉTODO

En esta sección se presenta el proceso de análisis para la construcción de *SistGeo*. Este proceso parte de la especificación de requerimientos presentada en la sección 2. A partir del cual se construyen el Modelo de Casos de Uso, el Modelo Objeto del Dominio y el Modelo de Interfaz; en la sección 5 se mostrará en detalle el proceso de prototipaje de la interfaz, haciendo énfasis en la utilización de los patrones de interacción.

Modelo de Casos de Uso

El Modelo de Casos de Uso permite identificar las funcionalidades del sistema a partir de las interacciones entre el usuario y el sistema. Describe la secuencia de acciones que produce un resultado observable al usuario.

En la figura 2 se presenta el diagrama de Casos de Uso de la aplicación, el cual muestra todos los escenarios posibles iniciados por un actor. Note que una secuencia de transacciones iniciadas por el usuario constituye un escenario de utilización del modelo. El conjunto de todos los escenarios muestra el comportamiento del Sistema.

Algunas relaciones entre los casos de uso de la figura 2 están dadas por la asociación «extend» que describe un curso alternativo opcional (la extensión) de otro caso de uso (base). Se usa cuando hay partes opcionales de un caso de uso, cuando hay cursos separados que son ejecutados en ciertas condiciones o cuando hay situaciones donde se puede seleccionar una, entre diferentes alternativas. Otras relaciones entre casos de uso están dadas por la relación «include», ésta se usa para particionar un caso de uso complejo en casos de usos constitutivos que son parte obligatoria del primero, permite separar una funcionalidad de un caso de uso, sea porque se repite en otros casos de uso o porque se quiere encapsular esa tarea.

A continuación se describen algunos casos de uso. El caso de uso principal -Crear Medio-, consiste en la creación de un medio geológico y, opcionalmente, en su recuperación si ya está creado -Cargar Medio-, así como seleccionar una de las diferentes posibilidades de tratamiento del medio -para modificarlo, para visualizarlo y para almacenarlo-. El punto de inclusión significa que debe existir un medio para manipularlo. El Caso de Uso Modificar medio abre las posibilidad de modificar una superficie o una línea de nodos, y una vez seleccionada la superficie, se selecciona la

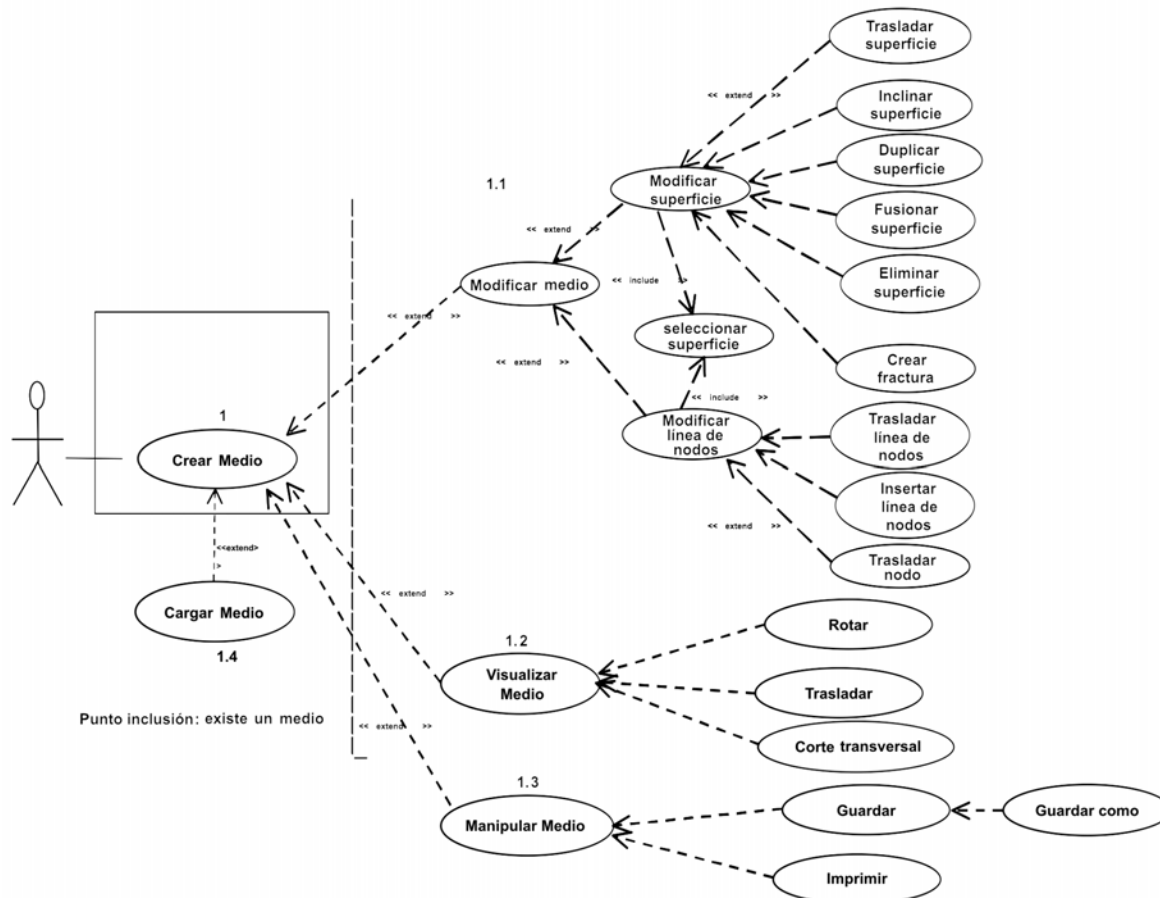


Figura 2. Modelo de Casos de Uso del sistema.

operación requerida. El Caso de Uso *Visualizar medio* reúne las facilidades de rotación, traslación, acercamiento, alejamiento y corte de un medio, para efectos de visualización. Finalmente, el Caso de Uso *Manipular Medio* agrupa las facilidades de manipulación de objetos típicas que permiten guardar, guardar como e imprimir.

Modelo Objeto del Dominio

El Modelo Objeto del Dominio muestra el conjunto de objetos que caracterizan el dominio del problema y las relaciones entre ellos.

Los objetos identificados en el dominio del problema se presentan en la figura 3, se muestran las clases conceptuales, agregando algunas de las operaciones asociadas a cada clase (los puntos suspensivos indican que existen otras operaciones, otras clases).

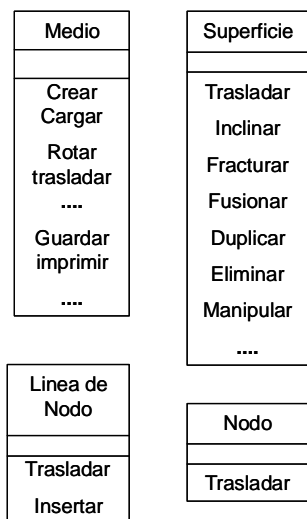


Figura 3. Clases conceptuales del modelo objeto del dominio.

Las clases identificadas son:

Medio: es la representación de una zona del subsuelo.

Región: es cada una de las partes en que se divide el medio.

Un medio está formado por una o más regiones. La unión de todas las regiones forma el medio. Las características de una región pueden ser establecidas de manera independiente de las otras regiones.

Superficie: es la frontera entre dos regiones del medio. Cada región está limitada por dos o más superficies.

Parche: es una superficie especial B-Spline rectangular paramétrica bivariable. Un parche queda completamente definido mediante su polígono de control. Cada parche posee características de continuidad C2 en toda su superficie.

Cuando una superficie se forma por la unión de dos o más parches, estos se conectan conservando en su frontera de unión continuidad geométrica G1. Así, una superficie posee continuidad geométrica G1 en los puntos de unión de dos parches y continuidad geométrica C2 en cualquier otro punto. Dos superficies pueden compartir uno o más parches.

Nodo: es cada uno de los puntos que forman el polígono de control de un parche.

Línea de Nodos: es cada una de las líneas de nodos que forman el polígono de control de un parche.

En la figura 4 se muestra el Modelo Objeto del Dominio, posteriormente se describen las relaciones entre las clases de objetos.

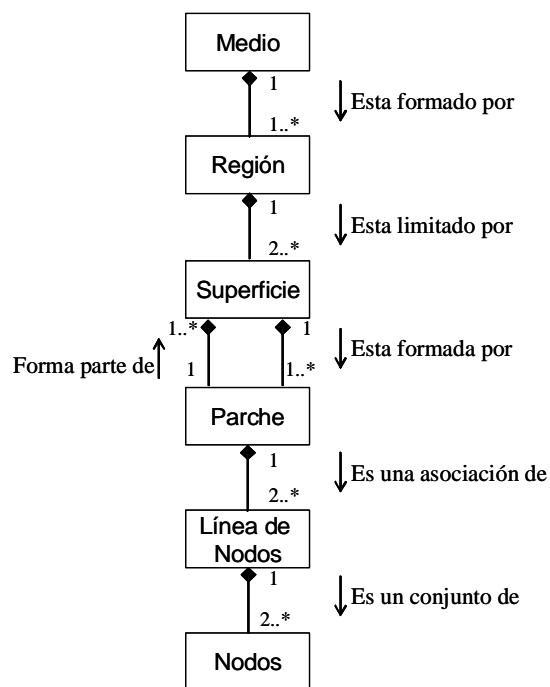


Figura 4. Diagrama del Modelo Objeto del Dominio.

Las relaciones entre estas clases son:

Relación «está formado por», existente entre las clases Medio y Región, indica que una instancia de medio está formada por una o más regiones, esto es, el medio se representa como un conjunto de regiones que constituyen una partición de una zona del subsuelo. Se utiliza el término Región, por ser más amplio que el concepto de capa. Si bien en el momento en que se crea el medio, todas las regiones tienen forma de capa, el sistema SistGeo provee operaciones que permiten modificar esta configuración inicial, de modo que puedan definirse regiones con formas arbitrarias.

- Relación «está limitada por», existente entre las clases Región y Superficie, indica que una región está conformada, inicialmente, por dos o más superficies o

interfases, una que la separa de la región inmediata superior y otra que la separa de la región que se encuentra por debajo; sin embargo, una región puede quedar limitada por más de dos superficies (o interfases), una vez aplicadas ciertas operaciones.

- Relación «está formada por», existente entre las clases Superficie y Parche, indica que cada superficie está formada por uno o más parches. Cuando se crea el medio cada superficie queda definida exactamente por un solo parche. Sin embargo, el sistema SistGeo proporciona algunas funcionalidades que permiten realizar operaciones sobre una superficie que implican un aumento del número de parches que la forman.
- Relación «forma parte de», existente entre las clases Parche y Superficie, indica que cada parche forma parte de, al menos, una superficie. Sin embargo, cuando una superficie está formada por más de un parche, es posible que ese parche sea compartido con una o más superficies, de modo que las superficies tengan este parche como sector común; en ese caso, el parche forma parte de todas las superficies que lo comparten.
- Relación «es una asociación de», existente entre las clases Parche y Línea de Nodos, indica que un parche puede entenderse como una malla compuesta de dos o más líneas de nodos (el polígono de control del parche) Para definir una malla se necesitan, al menos, dos líneas de nodos. La idea de agrupar los nodos en líneas de nodos permite definir operaciones más poderosas que aquellas que pueden definirse utilizando únicamente nodos individuales, aumentando así la capacidad de interacción del usuario.
- Relación «es un conjunto de», existente entre las clases Línea de Nodos y Nodo, indica que cada línea de nodos está formada por, al menos, dos nodos.

APLICACIÓN DE LOS RESULTADOS

Construcción del Modelo de Interfaz de Usuario

Construir el Modelo de Interfaz de Usuario significa establecer una colección de patrones (sea por reuso de patrones existentes o creación de nuevos patrones) y construir el Modelo Objeto de Interfaz, los cuales sirven de entrada para la construcción del prototipo de interfaz.

- Los Patrones de Interacción

Se utiliza el patrón de interacción para encapsular la información relativa a cada problema de interacción, identificando el problema, mostrando la solución y describiendo el contexto de esta solución; de esta forma se modela cada aspecto referente a la interfaz de usuario.

La colección de patrones de modelación de medios geológicos establece la base de un lenguaje de patrones en este dominio. Esta colección está conformada por diversos tipos de patrones. Estos patrones facilitan el desarrollo del Modelo Objeto de Interfaz y, en consecuencia, el prototipaje de la misma.

En esta sección se establece la colección de patrones de interacción aplicables al sistema. Se presenta, en la tabla 1, un meta-patrón para indicar el significado de cada componente del patrón. Algunos de estos componentes pueden obviarse, si no son requeridos en un determinado patrón, salvo los componentes: Nombre, Problema, Contexto y Solución, que son obligatorios.

Los patrones de interacción pueden encontrarse en los repositorios públicos de patrones, accesibles en el Web, como ejemplo se presentan las direcciones URL de Erickson y de Van Welie (consultadas en junio 2005).

A continuación se describen los tipos de patrones considerados en este trabajo, ello define una taxonomía de patrones:

- Patrones de Dominio: captura aspectos de interfaz correspondientes a sistemas del mismo tipo.
- Patrones del Sistema: están asociados al propósito del sistema, describen las metas de usabilidad que se desean alcanzar en el diseño de la interfaz de la aplicación.
- Patrones de Tareas: asociados a la representación de las funcionalidades en la interfaz. Por ejemplo, son los patrones asociados a las tareas de modelado de medios geológicos (crear un medio, traslación de superficie).
- Patrones de Elementos Compuestos: son patrones que describen estilos de interacción, tales como menús, formas, barras de herramientas, manipulación directa.
- Patrones de Elementos Simples: describen los componentes de más bajo nivel (botones, iconos, etc.).

Un lenguaje de patrones es una colección estructurada de patrones que guía al diseñador desde el problema a un alto nivel de abstracción a los problemas concretos.

A continuación se presenta la definición de un lenguaje de patrones (Acosta 2004), considerando la estructura del meta-patrón mostrado en la tabla 1:

- 1) Un *lenguaje de patrones* es un grafo acíclico dirigido $PL=(\emptyset,)$ con los nodos.
- 2) $\emptyset = \{P_1, \dots, P_n\}$ y los arcos $\mathfrak{R} = \{R_1, \dots, R_m\}$.

Tabla 1. Meta-patrón y Componentes

Componente	Descripción del componente del patrón
Nombre [clasificación]	El título del patrón, el cual debe permitir identificarlo fácilmente, comunicar su idea central y crear un vocabulario. La clasificación, que es opcional, indica el tipo de patrón.
Problema	Una descripción del problema desde el punto de vista del usuario.
Contexto	Una descripción de la situación en la cual puede usarse el patrón, cuáles son las características del contexto, en términos de las tareas, del usuario.
Solución	Una descripción de la solución, gráfica y textual - note que otros patrones pueden ser necesarios para completar la solución del problema.
Ejemplo	Una muestra ilustrativa de una solución exitosa donde se aplica el patrón.
Consecuencia	Describe los resultados de aplicar el patrón, mostrando la respuesta del sistema.
Justificación	Explicación de por qué la solución a ese problema es buena en el contexto. Describe cómo trabaja el patrón.
Impacto de usabilidad	Describe el impacto de usabilidad al aplicar el patrón y cuáles aspectos o indicadores de usabilidad son mejorados.
Fuerzas	Los conflictos que pueden afectar la solución para bien o para mal, aspectos del contexto que necesitan ser optimizados o las fortalezas.
Patrones Relacionados	Otros patrones con los que está relacionado, que completan la solución.

- 3) Cada nodo $P \in \wp$ representa un *patrón*.
 - p es el problema - s es la solución
- 4) Para dos patrones P y $Q \in \wp$, se dice que P *referencia* Q sí y sólo si existe un arco directo $R \in \mathfrak{R}$ que va de P a Q .
 - ctx es el contexto - f_p, \dots, f_i son las fuerzas
 - u es la usabilidad - con es la consecuencia
- 5) El conjunto de arcos que salen de un patrón $P \in \wp$ es denominado sus *referencias* y el conjunto de arcos que apuntan a este patrón es denominado su *ambiente*.
 - e_p, \dots, e_j son los ejemplos
 - ce_p, \dots, ce_m son los contraejemplos
- 6) Cada patrón $P \in \wp$ es, a su vez, una n-upla $P=(n, c, co, a, p, s, ctx, f_p, \dots, f_p, u, con, e_p, \dots, e_p, ce_p, \dots, ce_m, p_p, \dots, p_n)$ donde:

- n es el nombre - c es la clasificación
 - co es la confianza - a es el autor

Los patrones se identifican a diferentes niveles. Un patrón a un nivel puede implicar un número de patrones a un nivel más bajo para completarlo (ejemplo, un patrón de selección de operación puede implicar patrones para menús), esos patrones enlazados proveen una gramática informal para el

diseño: el lenguaje de patrones de un cierto dominio (Van Welie y Van der Veer, 2003). Un tratamiento exhaustivo de este lenguaje podría culminar en la definición formal del mismo.

A efectos de mostrar el método para la construcción del prototipo de la interfaz de usuario, se describen algunos de los patrones pertenecientes a la colección del sistema *SistGeo* (figuras 5, 6, 7 y 8). En cada patrón se presenta el diseño del primer borrador de cada pantalla. Así mismo, para cada patrón se indica su relación con el caso de uso correspondiente en la figura 2.

Es de hacer notar que para la construcción completa del prototipo de interfaz se definen y/o reusan otros patrones que, por razones de espacio, se omiten. Los patrones de elementos, generales a cualquier interfaz, tales como: caja de botones de opciones, botones simples, caja de giro, barra de herramientas -fijas y flotantes-, menús, ventana secundaria, entre otros; se encuentran en repositorios de patrones en la Web y pueden ser reusados.

La figura 9 muestra el diagrama que representa el lenguaje de patrones de *SistGeo* citado anteriormente, las flechas indican que el patrón origen referencia el (o los) patrón (es) destino; esto significa que el patrón origen requiere de las descripciones de los patrones destino para completar su descripción; por ejemplo para cargar un medio se requiere el patrón que describe la carga de archivo, utilizada usualmente.

- El Modelo Objeto de Interfaz

Para cada objeto se diseña su representación en la interfaz (reificación del objeto) y para cada operación activada por el usuario, el instrumento de interacción apropiado.

El Modelo Objeto de Interfaz incluye las representaciones de los objetos en la interfaz, producto de la reificación realizada a los objetos del Modelo del Dominio.

Es de hacer notar que para la construcción completa del prototipo de interfaz se definen y/o reusan otros patrones que, por razones de espacio, se omiten. Los patrones de elementos, generales a cualquier interfaz, tales como: caja de botones de opciones, botones simples, caja de giro, barra de herramientas -fijas y flotantes-, menús, ventana secundaria, entre otros; se encuentran en repositorios de patrones en la Web y pueden ser reusados.

La figura 9 muestra el diagrama que representa el lenguaje de patrones de *SistGeo* citado anteriormente, las flechas indican que el patrón origen referencia el (o los) patrón (es) destino; esto significa que el patrón origen requiere de las descripciones de los patrones destino para completar su descripción; por ejemplo para cargar un medio se requiere el patrón que describe la carga de archivo, utilizada usualmente.

- El Modelo Objeto de Interfaz

Para cada objeto se diseña su representación en la interfaz


Nombre y Tipo	Sistema de Modelado de Medios Geológicos - <i>SistGeo</i> (Patrón de Sistema)
Problema	El usuario requiere un sistema que le permita crear o cargar un medio existente para manipularlo.
Contexto	Inicio de la aplicación.
Solución	Iniciar el sistema desplegando la ventana en la cual se muestran las funcionalidades básicas al iniciar la aplicación: crear y cargar medio. 
Consecuencias	Una vez que el usuario ha seleccionado crear o cargar medio, se abre la ventana principal del sistema y la ventana secundaria correspondiente a la funcionalidad seleccionada.
Patrones Relacionados	Ventana de Manipulación del Medio, Crear Medio, Cargar Medio.

Figura 5. Ventana inicial del sistema de modelado de medios geológicos.

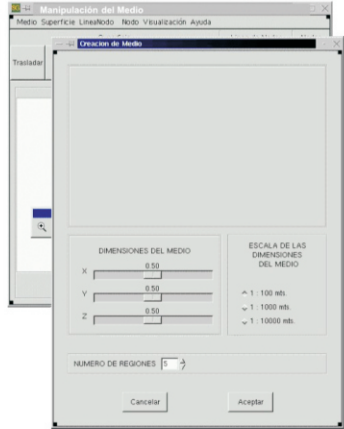
Nombre y Tipo	Creación de Medio. (Patrón de tarea)
Problema	El usuario desea crear un medio y visualizar la imagen tridimensional del medio en el espacio de trabajo
Contexto	El sistema se encuentra activo y el usuario ha seleccionado la funcionalidad Crear Medio
Solución	Se despliega la Ventana de Creación de Medio, la cual provee un área de visualización del medio a crear, y mecanismos de interacción para establecer: 1) las dimensiones del medio (el usuario selecciona en forma continua valores comprendidos dentro de un rango preestablecido, para cada una de las dimensiones), 2) la escala (este factor puede ser seleccionado mediante botones mutuamente excluyentes, que permiten seleccionar uno de un conjunto de factores de escala preestablecidos) y 3) el número de regiones que lo forman (selecciona un valor comprendido entre un valor mínimo hasta un valor máximo). Finalmente, el usuario acepta o cancela la operación de entrada de datos, presionando el botón correspondiente.
	
Consecuencias	Una vez que el usuario acepta la entrada de datos, se ejecuta la operación <i>Crear_medio</i> y en la Ventana Principal de Manipulación del Medio se despliega una imagen tridimensional del Medio creado y se habilitan los mecanismos que permiten acceder a las funcionalidades que actúan sobre el medio.
Patrones Relacionados	Los patrones relacionados son patrones de elementos: los correspondientes a Barra de desplazamiento, Caja de Giro y Botón y el patrón de tarea selección de operación.

Figura 6. Creación de medio (Caso de Uso Crear Medio).

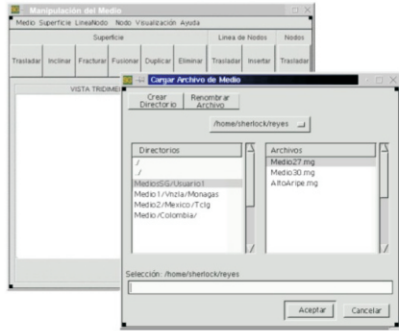
Nombre	Cargar Medio
Problema	El usuario desea cargar un medio ya almacenado
Contexto	El sistema se encuentra activo y el usuario ha seleccionado la funcionalidad Cargar Medio desde la ventana inicial del Sistema o desde la ventana principal de Manipulación del Medio.
Fuerzas	La selección del archivo de medio se encuentra restringida a un directorio particular y a los subdirectorios contenidos dentro de éste.
Consecuencias	Una vez que el usuario acepta la entrada de datos, se ejecuta la operación en la Ventana Principal de Manipulación del Medio y se despliega una imagen tridimensional del medio y se habilitan los mecanismos que permiten acceder a las funcionalidades sobre el medio.
Solución	Se despliega una Ventana de carga del medio (ventana estándar de carga de un archivo). El usuario debe especificar la ubicación (dispositivo de almacenamiento, directorio, nombre del archivo, etc.), donde se encuentra almacenado el medio.
	
Patrones Relacionados	Carga de archivo

Figura 7. Cargar un medio (Caso de Uso Cargar Medio).

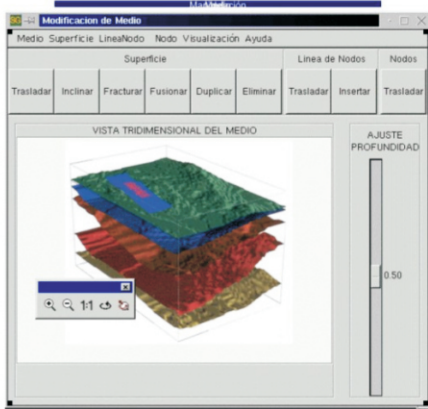
Nombre	Traslación de Superficie
Problema	El usuario desea trasladar una superficie interfase de un medio desplegado en el espacio de trabajo
Contexto	La Ventana Principal se encuentra activa y en ella se despliega una imagen tridimensional del medio de la cual se ha seleccionado una superficie.
Solución	<p>Para trasladar la superficie seleccionada el usuario debe activar la operación y establecer la magnitud del desplazamiento.</p> 
Fuerzas	El sistema implementa un mecanismo de detección de colisiones, que evita que se traslade la superficie más allá del límite donde se produce una colisión con otra superficie vecina.
Consecuencias	Al modificarse la posición de la barra de desplazamiento, se actualiza la imagen del medio con las modificaciones realizadas.
Patrones Relacionados	Los patrones selección de operación y selección de superficie

Figura 8. Traslación de Superficie (Caso de Uso Trasladar Superficie).

(reificación del objeto) y para cada operación activada por el usuario, el instrumento de interacción apropiado.

El Modelo Objeto de Interfaz incluye las representaciones de los objetos en la interfaz, producto de la reificación realizada a los objetos del Modelo del Dominio.

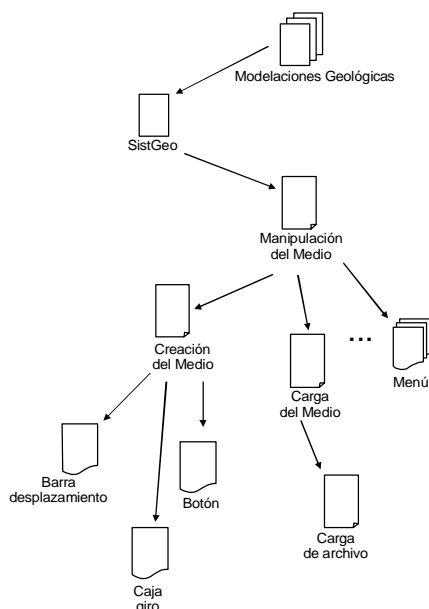


Figura 9. Parte de la Colección de Patrones de SistGeo.

Es de hacer notar que, debido a que la conformación de la Colección de Patrones (a través de la definición y/o reuso de estos) y la construcción del Modelo Objeto de Interfaz se realizan en forma paralela, es posible que puedan existir elementos que se describen en ambos modelos.

La figura 10 muestra un ejemplo de la aplicación del proceso de reificación a dos objetos del Modelo del Dominio, a saber el *Medio* y a la operación de carga de un medio ya almacenado.

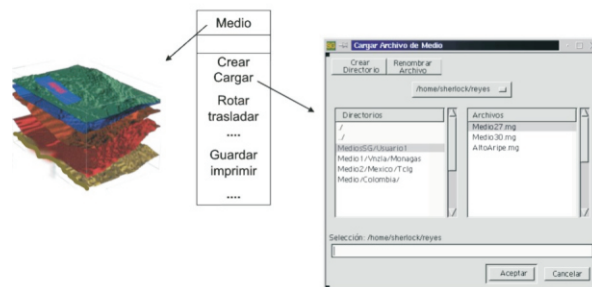


Figura 10. Ejemplo de reificación de objetos y operaciones.

- La construcción del prototipo de la interfaz de usuario

Con todos los artefactos de software obtenidos hasta el momento, se procede a la construcción del prototipo

ejecutable, para lo cual se selecciona una plataforma que facilite el proceso. Se implementan las soluciones descritas en los patrones, los objetos que conforman el Modelo Objeto de la Interfaz y la navegación entre las diferentes interfaces.

Una vez construido el prototipo, se continúa el desarrollo de la aplicación siguiendo cualquier metodología orientada a objeto, que incluya elementos actuales, tales como la utilización de componentes reusables, sea a nivel de diseño o de construcción.

CONCLUSIONES

El lenguaje de patrones de interacción conjuntamente con el modelo objeto de interfaz conforman el denominado Modelo de Interfaz de Usuario, el cual facilita la construcción del prototipo de interfaz de usuario desde las primeras etapas del desarrollo de software, específicamente, desde la etapa del análisis. Este trabajo ha permitido experimentar la factibilidad y efectividad de aplicar un método para la construcción de un prototipo de interfaz de usuario. Este método se ha definido de tal forma que puede ser incorporado a cualquier método de desarrollo de software orientado a objetos que permita la definición de, al menos, el Modelo de Casos de Uso y el Modelo Objeto del Dominio. En este caso de estudio, el método se incorporó sin problemas al Proceso Unificado descrito en Larman (2002), comprobando esa premisa. El dominio de este caso de estudio es bastante especializado al campo de la geología; ahora bien, las ideas planteadas pueden ser trasladadas a otros dominios con un rango un poco más amplio de aplicación y demanda actual como por ejemplo el Comercio Electrónico; generando así un lenguaje de patrones que quedaría disponible para su utilización en el desarrollo de cualquier aplicación de *e-commerce*. Esto se puede afirmar debido a que el método descrito es de propósitos generales y para abordar otro dominio sólo sería necesario cambiar el análisis de requerimientos de la nueva aplicación e incorporar especialistas del área respectiva.

Es de hacer notar que aunque en el equipo de diseño de la interfaz de usuario se incorporaron especialistas en el área del dominio, quienes aportaron sus ideas, observaciones y comentarios a fin de producir el prototipo presentado en este trabajo, este prototipo debe ser sometido a nuevas evaluaciones por parte de los usuarios finales del sistema, de tal manera que pueden surgir cambios en las interfaces descritas, producto de estas evaluaciones.

El hecho de haber incorporado especialistas del dominio de la aplicación, sin conocimientos profundos en el desarrollo de software, desde las primeras etapas del ciclo de vida de la aplicación permitió verificar el potencial de los patrones

de interacción como una herramienta que presenta una notación sencilla y fácil de aprender, siendo comprensible por todo el equipo multidisciplinario de desarrollo.

AGRADECIMIENTOS

Los autores de este trabajo hacen público su agradecimiento al Consejo de Desarrollo Científico y Humanístico (CDCH) de la Universidad Central de Venezuela y a la Universidad de Carabobo, por el financiamiento otorgado para la realización de este trabajo de investigación.

REFERENCIAS

- 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.
- BEAUDOUIN-LAFON, M. & MACKAY, W. (2000). Reification, polymorphism and reuse: three principles for designing visual interfaces. Proc. Advanced Visual Interfaces, AVI-2000, Palermo, Italy.
- ERICKSON, T. (2005). Patterns Home Page. URL: <http://www.visi.com/~snowfall> (Consultada Junio 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. USA.
- NIELSEN, J. (1993). Usability engineering. Prentice-Hall. USA.
- REYES, A. (1999). Algoritmo trazador de rayos sísmicos en medios heterogéneos débilmente anisotrópicos usando técnicas de optimización no lineal. (Tesis de Maestría). Postgrado en Ciencias de la Computación. Universidad Central de Venezuela. Caracas.
- RUMBAUGH, J., BOOCH, G. & JACOBSON, I. (2005). Unified Modeling Language User Guide. Addison-Wesley Professional, 2nd edition.
- VAN WELIE, M. (2005). Patterns Collection. URL: <http://www.welie.com> (Consultado Junio 2005).
- VAN WELIE, M. & VAN DER VEER, G. (2003). Pattern Languages in Interaction Design: Structure and Organization. Interact '03. Zurich. Eds. Rauterberg, Menozzi, Wesson p.527-534 IOS Press, Amsterdam. URL: <http://www.welie.com/> (Consultado junio 2005).