

Diagramas de Transición entre Interfaces de Usuario para la Especificación y el Diseño de Sistemas de Software.

M. Gómez, J. Cervantes

Departamento de Matemáticas Aplicadas y Sistemas, Universidad Autónoma Metropolitana
Artificios 40 Col. Hidalgo, Alvaro Obregón D.F., 01120 México.

RESUMEN

En este trabajo se introducen los *Diagramas de Transición entre Interfaces de Usuario* e ilustra, mediante un caso de estudio, cómo es que la etapa de diseño dentro del proceso de desarrollo de software se facilita considerablemente cuando el sistema queda especificado apoyándose en este tipo de diagramas. Describimos algunas experiencias positivas que hemos tenido con nuestros estudiantes y discutimos el impacto que creemos tendría el uso de dichos diagramas en la industria del Software.

Palabras clave: Diagramas de Transición entre Interfaces de Usuario, Modelado de Interfaces de Usuario.

1. INTRODUCCIÓN

El proceso de desarrollo de software se conoce como ciclo de vida del software porque describe la vida de un proyecto de software; primero nace con los requerimientos, luego se lleva a cabo su implantación, que consiste en su diseño, codificación y pruebas, posteriormente el producto se entrega, y sigue viviendo durante el mantenimiento. La vida de un proyecto de software termina cuando se deja de utilizar el sistema. Hay una gran variedad de modelos de desarrollo de software, en la mayoría de los cuales el sistema se desarrolla de forma evolutiva, es decir, el ciclo requerimientos-diseño-codificación-pruebas se repite para ir incrementando funcionalidades. Independientemente de cual sea el modelo de desarrollo de software que se adopte, todos y cada uno de ellos involucran las etapas de análisis de requerimientos y de diseño, las cuales son decisivas, ya que un requerimiento mal especificado o un mal diseño traerán como consecuencia un costo muy alto en la reparación de los defectos [1, 2, 3]. Las herramientas tales como UML son necesarias para que los procesos de especificación de requerimientos y de diseño sean actividades más disciplinadas y por lo tanto más efectivas.

En el caso ideal, los requerimientos están plenamente identificados y plasmados en un documento de *Especificación de Requerimientos* de manera precisa y clara antes de comenzar con el diseño, sin embargo, en la práctica esto no sucede así pues es difícil visualizar cuál es el sistema que se desea. Resulta más fácil comenzar con el diseño y, al avanzar en este, acabar de identificar, comprender y definir los casos de uso más complicados y replantearlos de tal manera que la *Especificación de Requerimientos* sea lo más clara posible. Este constante intercambio de información entre los procesos de análisis de requerimientos y de diseño indica el grado de dificultad que existe en lograr una *Especificación de Requerimientos* clara, completa y estable sobre la cual se puedan hacer estimaciones acertadas del resto del desarrollo del proyecto. Sería de gran utilidad contar con una herramienta que sirva de conexión entre los procesos de análisis de requerimientos y de diseño (de alto nivel) y que este intercambio de información se haga con un lenguaje común.

Por otra parte, cuando se quiere describir un sistema interactivo se tienen que tomar en cuenta dos componentes principales, la

interfaz de usuario, la cual permite la comunicación del usuario con el sistema, y las operaciones que éste ejecuta. La acción que ejecute el sistema está determinada por la información que el usuario proporciona a través de la Interfaz de Usuario (IU). El sistema puede responder de varias maneras: proporcionando resultados, solicitando información adicional, mostrando mensajes de error o mensajes de ayuda. Las Interfaces de Usuario juegan un papel fundamental y se debe prestar especial atención a su diseño desde las fases más tempranas del desarrollo del sistema. El diseño de las IU se puede llevar a cabo como la última parte del análisis de requerimientos cuando es el cliente quien determina sus características, o puede ser la primera fase del diseño cuando éstas quedan a criterio de los desarrolladores del sistema. Una tercera opción es que se desarrollen en colaboración entre cliente y desarrollador. Por esto es que resulta de suma importancia el contar con herramientas de diseño que permitan llevar a cabo esta actividad de la manera más eficaz posible.

En el presente trabajo explicamos cómo es que el uso de una nueva herramienta a la que llamamos *Diagramas de Transición entre Interfaces de Usuario* (DTIU) puede ayudar a facilitar el paso del análisis de requerimientos al diseño mejorando así la calidad del diseño. Los DTIU marcan una clara frontera entre la Especificación y el Diseño e identifican cual será la siguiente interfaz que se le presentará al usuario en función de la acción que lleve a cabo dentro de una cierta interfaz dada.

En la sección 2, se describen los Diagramas de Transición entre Interfaces de Usuario, en la sección 3 se muestra su aplicación y su utilidad mediante un caso de estudio. En la sección 4 se mencionan algunas experiencias con el uso de los DTIU y finalmente en la sección 5 se comparan los DTIU con otros Diagramas UML.

2. LOS DIAGRAMAS DE TRANSICIÓN ENTRE INTERFACES DE USUARIO (DTIU).

Los artefactos para modelar sistemas de software se clasifican en dos vertientes: los modelos para el desarrollo orientado objetos y los modelos para el desarrollo estructurado de sistemas. El Lenguaje Unificado de Modelado (UML) [4] es un lenguaje estándar para modelar aplicaciones software con orientación a objetos. UML contiene diagramas para modelar globalmente los elementos de un sistema y sus relaciones entre sí (vista estática), como lo son los diagramas de clases, de componentes y de objetos, también incluye diagramas para modelar el comportamiento del sistema a lo largo del tiempo (vista dinámica) como los diagramas de casos de uso, de máquina de estado, de actividades y de secuencias.

También existen herramientas gráficas para el modelado de sistemas estructurados, éstas son: los Diagramas de Entidad-Relación (DER), los Diagramas de Flujo de Datos (DFD) y los Diagramas de Transición de Estados (DTE), cada uno de ellos brinda una visión diferente del sistema. El primero pone énfasis en los datos y sus relaciones, el segundo centra la atención en la funcionalidad del sistema y el tercero en el comportamiento dependiente del tiempo. El Diccionario de Datos (DD) es un complemento a estas herramientas el cual nos permite definir con un mayor grado de detalle los datos presentes en los diagramas [1, 5, 6].

Los *Diagramas de Transición entre Interfaces de Usuario* (DTIU) son una solución a la carencia de una herramienta única que modele el comportamiento del sistema a través de las Interfaces de Usuario. Hasta ahora, las IU se han modelado combinando varios diagramas UML, por ejemplo, da Silva & Paton [7] lo hicieron combinando diagramas de clases, diagramas de actividades y diagramas de secuencia, sin embargo no siempre es claro como modelarlas. Phillips & Kemp [8] propusieron dos tipos de tablas para complementar la información faltante en los diagramas UML. Da Silva & Paton identificaron que los problemas principales para modelar Interfaces de Usuario con UML son los siguientes: “UML no describe claramente la relación entre los casos de uso y las actividades, tampoco contiene una notación para describir las representaciones abstractas de una interfaz de usuario y además es difícil identificar como se relaciona una actividad que implique la interacción del usuario con una Interfaz de Usuario determinada” [7]. La principal ventaja de los DTIU con respecto a esta propuesta es que muestran los detalles de interés para el cliente en un solo diagrama, y no incluyen detalles de implementación, sin embargo proporcionan a los diseñadores suficiente información como para poder comenzar a implantar las tareas solicitadas. Traetteberg, en su tesis doctoral [9] aborda el problema de encontrar modelos que puedan integrar la representación de las interfaces de usuario en el desarrollo de sistemas interactivos, en esta tesis se concluye que los modelos que se deben usar para el diseño de las IU están determinados por las características específicas y las necesidades del diseño. Los DTIU pueden sumarse a esta propuesta. El DTIU es una vista especializada de una máquina de estados con un formato particular, el cual se describe a continuación.

Un DTIU consiste en un pseudografo dirigido cuyos vértices (rectángulos) representan cada uno a una y sólo una interfaz de usuario y sus aristas (flechas con una etiqueta sombreada) representan las actividades que el usuario realiza al interactuar con el sistema (véase un ejemplo en la *Figura 1*). Cada flecha está etiquetada por medio de un óvalo con el nombre de una actividad. Las actividades contienen en su descripción verbos que el usuario ejecuta. Cada interfaz de usuario tiene un nombre y un número que la identifica (rectángulo del lado izquierdo). Por separado se debe ilustrar el diseño de cada una de estas interfaces.

La interpretación de un DTIU es la siguiente: Cada vez que el usuario inicia una actividad se produce una transición desde la interfaz en donde se inició (origen) hasta otra interfaz (destino) como resultado de tal actividad. Una actividad puede llevar a otra interfaz en la que se pueden realizar a su vez otro conjunto de actividades o bien, puede dejar al usuario en la misma interfaz.

Para que sea efectivo, el diagrama debe tener la profundidad suficiente como para describir todos los casos de uso del sistema en los que interviene el usuario. Normalmente, por la complejidad y tamaño del diagrama, éste se presenta por secciones. Con el DTIU el sistema debe quedar totalmente descrito a nivel de funcionalidad para el usuario. En el DTIU se describen únicamente las interfaces con el usuario. Las interfaces con bases de datos o las de comunicación con otros sistemas pueden describirse en los Diagramas de Secuencias, y/o en los Diagramas de Colaboración de UML.

Un DTIU contiene el nombre de cada una de las interfaces de usuario y un número que permitirá identificarlas fácilmente durante el diseño y etapas posteriores. El objetivo de un DTIU es plasmar gráficamente todos los servicios que el usuario puede pedir al sistema de una manera sencilla para que lo puedan interpretar tanto el diseñador como el cliente.

3. CASO DE ESTUDIO: QUALITEAM

Un problema importante en la ingeniería de software es que existe una amplia gama de métodos, lenguajes, herramientas y ambientes

cuya utilidad no se ha mostrado con ejemplos prácticos [10]. A continuación ilustramos, mediante un caso de estudio, la aplicación de los *Diagramas de Transición entre Interfaces de Usuario* como parte de la metodología de la especificación del “Sistema Gestor de Proyectos de Software: QualiTeam” [11] el cual es un sistema que tiene como objetivo facilitar tanto la administración y seguimiento de proyectos de software por parte del líder de proyecto (el profesor) como el entendimiento de conceptos de aseguramiento de la calidad y de ingeniería de software por parte de los diseñadores (estudiantes). El sistema brinda al profesor control y a los alumnos una guía sobre las actividades de documentación que se deben llevar a cabo a lo largo de un proyecto de desarrollo de software. Únicamente se presentan las funcionalidades de “seguridad” y de “administración de proyectos”. Por razones de espacio no se incluye el detalle de “Requerimientos de Cambio”, “Pruebas”, “Reportes de Error” ni de “Inspecciones”.

3.1 DTIU en la Especificación de Requerimientos.

Un DTIU es de gran ayuda para especificar las acciones del usuario en el sistema. Por ejemplo, el DTIU de la *Figura 1* muestra la funcionalidad de *seguridad*, desde el punto de vista del usuario. Proponemos que cada actividad en un DTIU produzca una y sólo una transición hacia alguna de las interfaces y que cuando sea necesario se identifiquen más actividades distintas. Por ejemplo, podría pensarse en una sola actividad para *introducir UserId y Password* y que la transición correspondiente dependa de si estos datos son correctos o no, lo que implicaría un elemento condicional o bifurcación en el diagrama. Pensamos que esto no es bueno ya que al cliente se le presentarían elementos relacionados con la implementación y no con los aspectos que a él o ella le interesan.

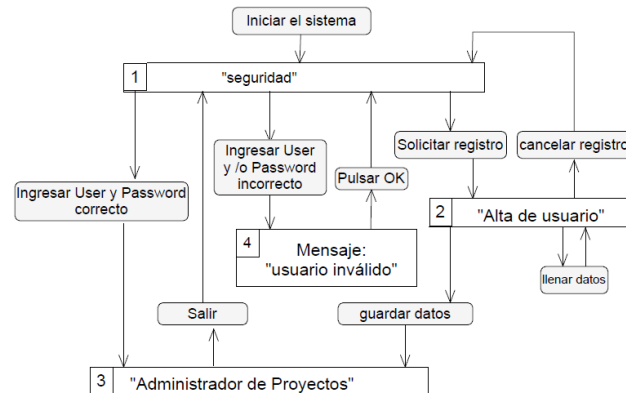


Figura 1 DTIU del subsistema de seguridad.

Si se considera que introducir UserId y Password correctamente o incorrectamente son dos actividades distintas, cada una tendrá su correspondiente transición, esto queda suficientemente claro para el cliente. El diseñador podrá resolver posteriormente la manera en la que el sistema distinguirá entre estas dos actividades. En la *Figura 1* se puede apreciar como el sistema envía un mensaje de error y regresa a la interfaz original cuando el usuario proporciona UserId y/o Password incorrecto mientras que en el otro caso avanza hacia la interfaz principal del sistema.

En la *Figura 2* se pueden apreciar las acciones que el usuario puede llevar a cabo desde la interfaz #3. Para el diseñador es claro en el DTIU que, si se elige “Dar de alta un nuevo proyecto” el sistema hará un conjunto de tareas diferentes a las que se deberán llevar a cabo en el caso de que el usuario elija “Editar proyecto”, aunque al usuario se le muestre la misma interfaz #5 “Edición del proyecto”. Las interfaces #3 y #5 están ilustradas en las *Figura 6* y *Figura 7* respectivamente.

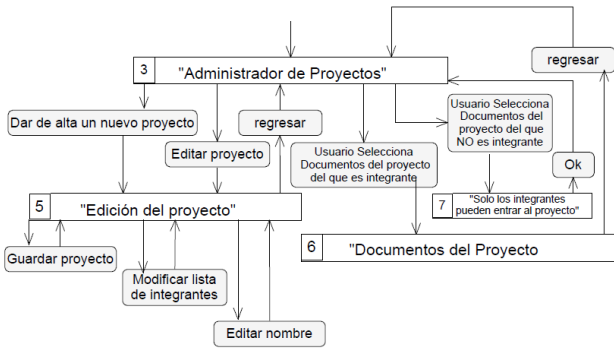


Figura 2 DTIU desde la interfaz “Administrador de Proyectos”.

En el DTIU de la Figura 3 queda documentada la forma en la que el usuario, a través de la interfaz #6, puede acceder a los 4 diferentes subsistemas de QualiTeam (interfaces #14, #15, #16 y #17). La acción de *ver detalle del documento* lleva a la interfaz #8 y la *crear nuevo documento* a la #12. En las Figura 4 y Figura 5, se muestran todas las actividades que puede hacer el usuario al “Editar las propiedades” y al “Ver Detalle” de un documento.

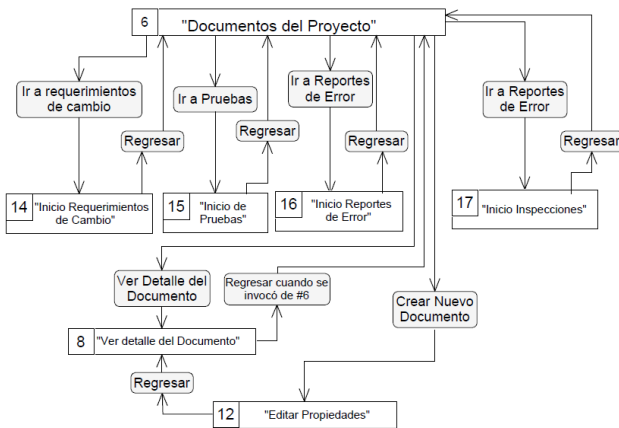


Figura 3 DTIU desde la interfaz “Documentos del Proyecto”.

Las acciones de “Editar Propiedades”, “Crear Nuevo Documento” y “Ver Detalle del Documento”, son acciones que se llevan a cabo también desde cada uno de las interfaces de las funcionalidades de: “Requerimientos de Cambio”, “Pruebas”, “Reportes de Error” e “Inspecciones”, por lo tanto, la opción de “Regresar” desde la interfaz # 8 depende del subsistema del cual se haya invocado. Lo anterior queda expresado en el DTIU al especificar que la actividad es *Regresar cuando se invocó desde la interfaz #x*, donde *x* representa la interfaz de origen, en el caso de la Figura 3 el regreso es hacia la interfaz #6.

Es importante hacer notar que poner el nombre y un número de identificación a cada interfaz de usuario dentro de los *Diagramas de Transición entre Interfaces de Usuario* permite una buena organización durante el diseño ya que se hará referencia a cada una de estas interfaces en esta y en todas las etapas posteriores.

Un DTIU, como Diagrama de Transición de Estados, enfatiza la secuencia temporal de los eventos del usuario y permite modelar en conjunto los eventos que no tienen un orden preestablecido.

Los casos de uso capturan los requerimientos funcionales del sistema y los expresan desde el punto de vista del usuario. Un *caso de uso* [4, 12, 13] especifica el comportamiento de una parte del sistema mediante una secuencia de interacciones entre los *actores* y el sistema. Los *actores* pueden ser personas, otro sistema de software

o un sistema de hardware. Los diagramas de casos de uso muestran las relaciones entre los servicios de un sistema y sus actores. Los diagramas de casos de uso dan sólo una visión general ya que la gran parte de la información se encuentra en la *descripción del caso de uso* [13].

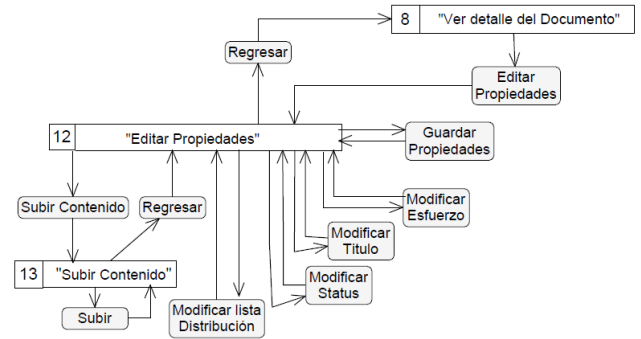


Figura 4 DTIU desde la interfaz “Editar Propiedades del Documento”

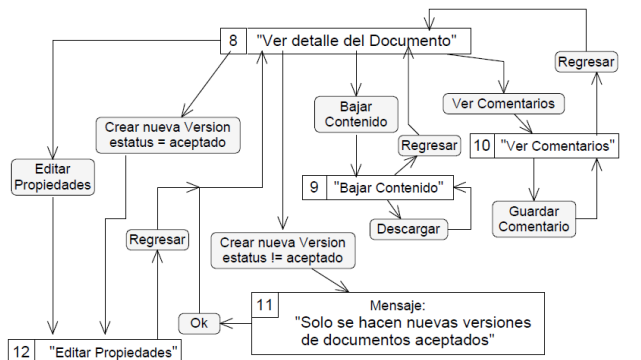


Figura 5 DTIU desde la interfaz “Ver Detalle del Documento”.

Los proyectos de software pueden agruparse en varios tipos. Wiegars [14] los clasifica en: aplicaciones interactivas para el usuario final, aplicaciones computacionalmente intensivas, almacenamiento de datos, procesamiento por lotes y productos hardware con software embebido. Los proyectos de aplicaciones interactivas para usuarios, incluyendo aplicaciones web y kioscos de ventas, son los que pueden especificarse mediante casos de uso y son precisamente para este tipo de proyectos para los que son útiles los DTIU.

Una de las ventajas de los DTIU es que pueden incluirse no solamente los casos de uso exitosos, sino también puede expresarse la manera en la que el sistema reacciona cuando el usuario lleva a cabo acciones inusuales o fuera de la secuencia normal. Por ejemplo, uno de los requerimientos particulares de QualiTeam es que solo se hagan nuevas versiones de documentos que estén en el estado de *aceptado*. En la Figura 5 se muestra la forma de expresar en un DTIU lo que sucede si un usuario pretende hacer la nueva versión de un documento que no está *aceptado*. Para este ejemplo, el sistema no permite elaborar una nueva versión y muestra la interfaz #11 con el mensaje de error “Sólo se crean nuevas versiones de documentos aceptados”, cuando el usuario pulsa en el botón *OK* el sistema se regresa a la interfaz #8 *Ver Detalle*. De esta forma, las condiciones que pueden ser previstas desde la etapa de análisis de requerimientos se documentan como una actividad más en el DTIU, ya sea exitosa o fallida, además, el mensaje de error que se mostrará en el caso no exitoso, queda claramente documentado.

Los *casos de prueba* constan de una serie de entradas, condiciones de ejecución y resultados esperados para verificar que se cumple con un requerimiento específico [15]. Otra de las ventajas de

los DTIU es que, además de los casos de uso, en ellos están identificados los *casos de prueba a nivel de sistema* en los que interviene alguna interfaz de usuario.

3.2 DTIU para el diseño de Interfaces de Usuario.

El diseño de las interfaces de usuario debería hacerse al mismo tiempo que los DTIU. Gran parte de los diagramas de secuencia, que se elaborarán después de que se definan los módulos en los que se dividirá el sistema, se basará en el diseño de los DTIU.

El DTIU resulta una herramienta muy útil para definir los elementos que deben estar presentes en cada interfaz. Se debe revisar constantemente la consistencia entre el DTIU y estos elementos. Por ejemplo, en la interfaz #3 de la *Figura 6* se reflejan todas las actividades documentadas en los DTIU de la *Figura 1* (salir) y de la *Figura 2* (Crear nuevo proyecto, editarlo, consultar los documentos del proyecto).

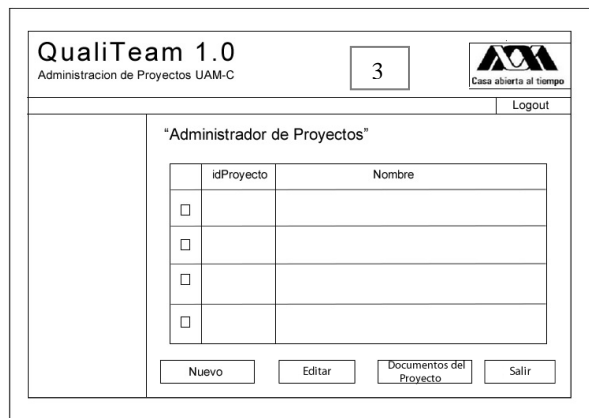


Figura 6 Interfaz de usuario #3: Administrador de proyectos.

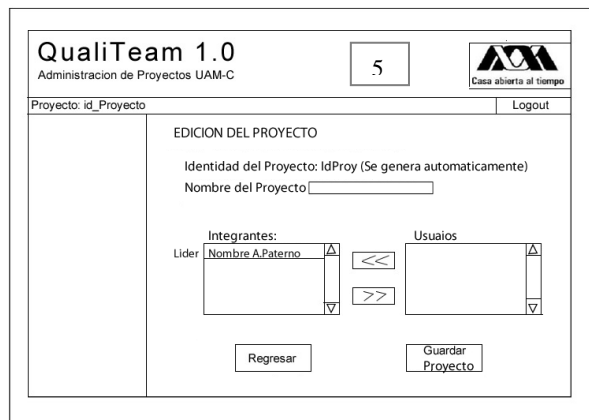


Figura 7 Interfaz de usuario #5: Edición del proyecto.

El bosquejo de las interfaces de usuario juega un papel fundamental en el diseño, estos bosquejos se utilizan constantemente durante el diseño de las interfaces. Durante esta actividad saltan a la vista defectos frecuentes como por ejemplo: la falta de una opción para regresar a la pantalla anterior o de un botón para guardar los datos capturados. Estos bosquejos cambian constantemente al

principio del diseño y cuando finalmente quedan estables ahorran mucho tiempo a los desarrolladores, ya que no es lo mismo cambiar el dibujo en un papel, que hacer cambios estructurales en una interfaz real. La codificación y compilación de las interfaces reales quedan minimizadas cuando se implementa solamente el diseño definitivo una vez que éste haya sido aprobado.

En las *Figura 6* y *7* se muestran los diseños de algunas de las interfaces de usuario del “Administrador de Proyectos” de QualiTeam en ellas se observa que se pueden realizar todas las acciones indicadas en el DTIU de la *Figura 2*.

Como se puede observar en estos ejemplos, diseñar lo que sucederá en el sistema cuando se pueden visualizar las interfaces de usuario es mucho más sencillo que cuando todo queda a la imaginación del diseñador. Con estos bosquejos se pueden identificar algunos errores y “cabos sueltos” sin tener que codificar y probar el sistema, además se pueden hacer varios diseños hasta encontrar el óptimo y modificar las UIs cuantas veces sea necesario, algunas modificaciones podrían afectar a los DTIU.

3.3 DTIU en el diseño arquitectónico del sistema.

Una vez que se tiene el *Diagrama de Transición entre Interfaces de Usuario de Usuario*, se definen los módulos en los que se sub dividirá el sistema. Cada actividad del usuario descrita en los DTIU implica una serie de tareas que el sistema debe realizar. Estas tareas se agrupan y se definen así los módulos del sistema. La idea es agrupar las tareas similares en un solo módulo. Algunas tareas, por ser muy similares, podrán ser unificadas en un módulo de servicio para evitar codificarlas en diferentes módulos promoviendo así la *reutilización*. Para nuestro caso de estudio, identificamos en el DTIU completo del sistema (no ilustrado) diversas actividades que invocan a las interfaces #8 y #12 desde las interfaces correspondientes a diferentes funcionalidades del sistema (#6, #14, #15, #16 y #17) lo que sugiere la reutilización de su código. Estas tareas están relacionadas con la gestión de documentos por lo tanto identificamos los subsistemas: “seguridad”, “administración de proyectos”, “Requerimientos de Cambio”, “Pruebas”, “Reportes de Error”, “Inspecciones” y un módulo de servicio “Gestor de Documentos”.

El subsistema *Gestor de Documentos* es un módulo que da servicio a los otros módulos, su tarea principal es la de administrar todo lo relacionado con los documentos en forma genérica como la creación, edición y consulta, así como el sistema de comentarios sobre los documentos. No hay una funcionalidad explícita representativa del módulo *Gestor de Documentos* en los DTIU, sin embargo, en estos diagramas se tiene una visión en conjunto de las actividades y las transiciones, con lo que creemos resulta más fácil identificar las tareas comunes.

3.3 DTIU en el diseño de las interacciones entre los módulos.

Sostenemos en este trabajo que los DTIU ayudan también a la tarea de la elaboración de diagramas de secuencia cuando ya se han definido los módulos del sistema. Los *diagramas de secuencia* son la herramienta indicada para diseñar las interfaces entre los módulos. Es necesario establecer un diagrama por cada *caso de uso* en los que se verá reflejada cada una de las transiciones en el DTIU.

Una de las ventajas de utilizar el DTIU al comienzo el diseño, es que en función de éste se pueden definir los diagramas de secuencia. Como todas las actividades del usuario están contenidas en el DTIU, se pueden diseñar todos los diagramas de secuencia necesarios tomando como base las actividades del usuario en el DTIU. Cada diagrama de secuencia se limitará a describir como se comunican los módulos del sistema desde que se inicia una actividad en una IU hasta que se realiza la transición hacia la misma o hacia otra IU. Para diseñar un diagrama de secuencia se incluyen los módulos que

intervendrán en una transición a tratar. Se parte de una interfaz de usuario inicial y de una actividad que ejecuta el usuario a través de esa interfaz, el diagrama culmina cuando se presenta al usuario la siguiente interfaz. Entre estos dos eventos se definen en secuencia todas las interacciones previstas entre los módulos del sistema. Al tomar como base los DTIU, todas las actividades del usuario quedarán previstas y, por lo tanto, todos los servicios que éste espera del sistema quedarán detallados en los diagramas de secuencia.

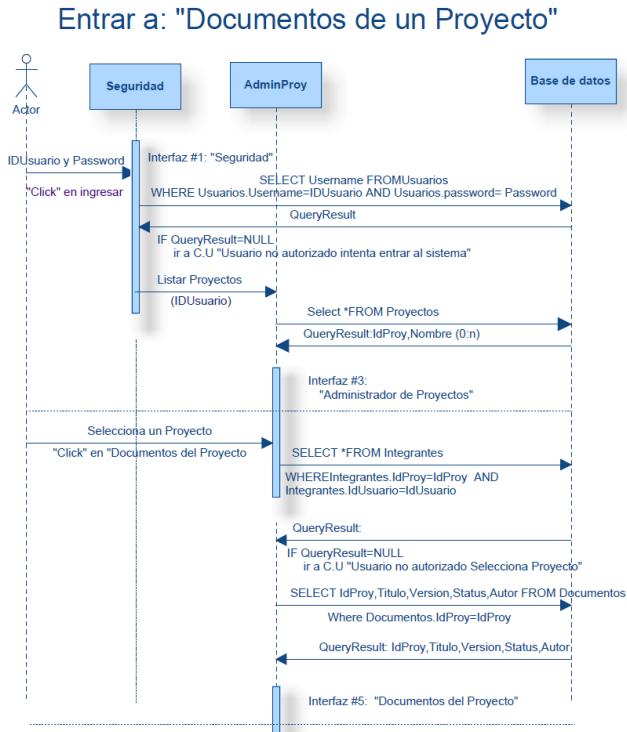


Figura 8 Diagrama de secuencia del caso de uso: "Entrar a los documentos de un proyecto".

Para hacer los diagramas de secuencia más claros, es mejor considerar primero si hay una condición anormal, y en caso de que así sea, se transfiere el control a un caso de uso no exitoso o excepcional que contemple esta situación, de otra forma se prosigue con la secuencia normal, de esta manera se evita la confusión gráfica que provocan los "else". Lo anterior se ilustra con el ejemplo de la Figura 8, en la que se presenta el diagrama de secuencia para el caso de uso "Entrar a los Documentos de un proyecto". Nótese que cada fin de transición está marcado con una línea punteada horizontal.

4 ALGUNAS EXPERIENCIAS CON EL USO DE LOS DTIU

Durante la Especificación y el Diseño de QualiTeam intervinieron alrededor de diez estudiantes de Ingeniería en Computación de la UAM-Cuajimalpa. En un principio se comenzó a diseñar sin los DTIU y a la gran mayoría se le dificultaba comprender los diagramas de secuencia y comprender qué es exactamente lo que el usuario puede hacer en el sistema. Además hubo ambigüedades y confusiones cuando intentaron modelar el sistema con UML pues para algunos no es claro qué herramientas deben usar para la Especificación de Requerimientos y cuales usar para el Diseño. Por ejemplo, hubo quien quiso modelar el sistema mediante diagramas de clases pero el resultado fue un diagrama demasiado complicado que no daba información útil para tener una visión global del sistema e incluía demasiado detalle. Al usar los DTIU los alumnos pudieron distinguir los requerimientos con mayor facilidad y así continuar con el diseño. La interacción del usuario con el sistema quedó

completamente especificada con los DTIU y, a su vez, los DTIU sirvieron de punto de partida para el Diseño. Además, la numeración de las interfaces facilitó la comprensión de los diagramas de secuencias que se hicieron posteriormente y permitió aclarar a los alumnos que aunque dos interfaces sean muy parecidas y tengan solo diferencias sutiles son necesarias para distintos casos de uso.

La claridad con la que queda especificado el diseño de interfaces de usuario en el DTIU ayudó mucho a nivel académico, sin embargo creemos que también será útil para diseñadores y desarrolladores de sistemas de software.

5 DISCUSIÓN: COMPARANDO LOS DTIU CON OTROS DIAGRAMAS.

Los Diagramas de Transición de Estados (DTE) de UML descritas por Booch et al. [4], describen el comportamiento de un sistema dirigido por eventos y modelan las posibles historias de vida de un objeto. Cuando ocurre un evento se puede generar una transición de estado la cual puede incluir o no una acción del sistema. Los DTE se pueden utilizar para describir interfaces de usuario, sin embargo también se utilizan para describir controladores de dispositivos y otros sistemas reactivos [4], por lo que su notación no es tan especializada como la de los DTIU y por lo tanto no se pueden incluir en ellos los mismos detalles que se incluyen en los DTIU.

El Diagrama de Actividades (DA) de UML es una forma especial del DTE. Presenta el orden y distribución de las actividades relacionadas a una tarea en particular condicionada por nodos de control. En los DA es posible modelar actividades del usuario y del sistema, sin embargo no se puede precisar cuál es la interfaz de usuario en la que se lleva a cabo cada actividad.

Un Diagrama de Componentes muestra la dependencia de las Interfaces de Usuario con otros componentes de un sistema, no está pensado para modelar el comportamiento de las Interfaces de Usuario (IU), aunque algunos de sus componentes pueden ser IU, solamente se indica la interacción de éstas con otros componentes. Los Diagramas de Interacción de UML, como los Diagramas de Secuencia o los Diagramas de Colaboración, muestran la interacción concreta entre un conjunto de objetos y los mensajes entre ellos, estos tres diagramas mencionados incluyen detalles de diseño sin interés para el cliente.

Los Diagramas de Secuencia se utilizan ampliamente para especificar requerimientos porque describen interacciones concretas y por lo tanto es fácil que los entiendan tanto expertos como clientes [16]. Sin embargo, no son muy útiles para especificar las acciones que el usuario puede realizar a través de las Interfaces, además con este tipo de diagramas no se pueden representar la perspectiva global del producto.

Para hacer un modelado equivalente al DTIU con UML se requiere usar varios diagramas y relacionarlos entre sí, por ejemplo, da Silva & Paton [7] se basan en los Diagramas de Clases y de Casos de Uso. Los Diagramas de Casos de Uso fueron diseñados para el análisis de requerimientos, por lo que no proporcionan información del flujo de las tareas, necesario para la implementación, para incluir esta información, usan como complemento Diagramas de Actividades y Diagramas de Secuencias. Como este modelado contiene inclusive las clases de las interfaces, el detalle es tal, que se pierde la visibilidad, especialmente para el cliente.

Según da Silva & Paton, "UML tiene un amplio conjunto de constructores lo suficientemente completo como para modelar los aspectos arquitectónicos de interfaces de usuario basadas en formularios" [7], sin embargo, falta una herramienta que permita modelarlas desde una perspectiva global para los diseñadores y detallada para el cliente. La información en los DTIU le dan al cliente una idea clara de los servicios que el sistema le proporcionará

y al diseñador le permite asociar cada servicio con una interfaz específica desde la cual se inician las acciones, el diseñador decidirá la forma en la que el sistema cumplirá con el servicio.

Las acciones del usuario que actúan sobre una misma interfaz, por ejemplo, aquellas que actualizan información, relocalizan o envían a secciones dentro de la misma, están fuera del alcance de los DTIU. Este tipo de detalles normalmente son decisiones de diseño para mejorar la usabilidad del sistema y no forman parte del análisis de requerimiento.

En los DTIU se aprecian las interfaces de usuario que intervienen para una determinada secuencia de actividades y tiene dos ventajas fundamentales: 1) ayuda a que el cliente y el analista lleguen a un acuerdo sobre las tareas que llevará a cabo el usuario en el sistema y, 2) mejora el entendimiento por parte de los diseñadores, de lo que el analista específica. No existe un diagrama UML con un propósito así de especializado.

6 CONCLUSIONES

La introducción del *Diagrama de Transición entre Interfases de Usuario* permite identificar todo lo que puede hacer un usuario en el sistema y relacionarlo con la interfaz que se le presentará. El DTIU permite describir totalmente a un sistema en base a las acciones que un usuario puede llevar a cabo y a la forma en que el sistema debe responder a cada una de estas acciones. De esta manera, complementa la *Especificación de Requerimientos*. El DTIU constituye un enlace natural entre las etapas de Análisis de requerimientos y de Diseño. Además, facilita en gran medida el bosquejo de las interfaces de usuario y el resto del diseño. Por lo anterior, recomendamos que una *Especificación de Requerimientos* incluya los *Diagramas de Transición entre Interfases de Usuario*. De esta manera se resuelve el problema identificado por Berzal [17] sobre la carencia en UML de una herramienta que ayude con el modelado de las interfaces de usuario.

Los DTIU son útiles tanto para sistemas pequeños como para los muy grandes. El uso de esta herramienta puede llegar a tener un impacto importante en la industria del software, ya que ataca un problema medular: la falta de entendimiento entre los que solicitan un sistema y los que lo van a implantar.

7 TRABAJO FUTURO

Estamos aplicando los DTIU en el desarrollo de varios proyectos y hemos encontrado casos no previstos que pueden ser resueltos favorablemente, como es el caso de elementos que aparecen dentro de una Interfaz de Usuario solamente cuando se cumple cierta condición.

8 REFERENCIAS

- [1] Pfleeger, Shari L. "Ingeniería de software. Teoría y práctica" 1ª edición. Editorial Pearson Education, Buenos Aires, 2002.
- [2] Sommerville, I.: "Ingeniería del Software", 7ª Ed., Pearson Addison Wesley, Madrid, 2005.
- [3] Weitzenfeld, A.: "Ingeniería de Software Orientada a Objetos con UML, Java e Internet". Editorial Thomson, 2004.
- [4] Booch G., Rumbaugh J, & Jacobson I., "The Unified Modeling Language User Guide". Addison-Wesley, Reading, MA, 1999.
- [5] Pressman, Roger S. "Ingeniería del Software: Un enfoque práctico", 5a edición. Editorial McGraw Hill, España, 2002.
- [6] Piattini M., Calvo-Manzano J., Cervera J., Fernandez L. "Análisis y diseño de Aplicaciones Informáticas de Gestión". Una perspectiva de Ingeniería de Software. Alfaomega-Rama, México, 2004.
- [7] da Silva P. & Paton N., "User Interface Modelling with UML", IEEE Software, Vol.20, No 4, 2003.

- [8] Phillips, C. and Kemp, E., "In Support of User Interface Design in the Rational Unified Process". J. Australian Computer Science Communications, Vol 24, No. 4, pp. 21-27, 2002.
- [9] Traetteberg, H., "Model-based User Interface Design". Ph. D. Thesis, Norwegian University of Science and Technology, 2002.
- [10] García, F., Serrano M., Cruz-Lemuz J., Genero M., Calero C. y Piattini M., "Empirical Studies in Software Engineering Courses: Some Pedagogical Experiences", International Journal of Engineering Education, Vol. 24, No. 4, pp. 761-771 2008.
- [11] Cervantes J., Consuelos J:C. y Gómez M., "QualiTeam: Sistema Gestor de Proyectos de Software". Teorías, Modelos y Aplicaciones de Matemáticas y Computación". Memorias de la 3ª y 4ª Semana de Computación y Matemáticas Aplicadas SCMA'10, SCMA'11. Universidad Autónoma Metropolitana Unidad Cuajimalpa (in press).
- [12] Ceria, S., "Casos de Uso, un método práctico para explorar requerimientos". Universidad de Buenos Aires. http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf
- [13] Vega, M., "Casos de Uso UML", Universidad de Granada. <http://lsi.ugr.es/~mvega/docis/casos%20de%20uso.pdf>, 2010.
- [14] Wiegers, K., "More about software requirements. Microsoft Press, 2006.
- [15] IEEE-829 "Standard for Software and System Test Documentation". IEEE Computer Society, 2008.
- [16] Li, X., Qiu L., X, Wang X., Chen Z., Zhou L. Yu J. Zhao. "UML interaction model-driven runtime verification of Java programs". IET Softw., Vol 5, No 2, p.142-156 doi:10.1049/iet-sen.2009.0009, 2011.
- [17] Berzal, F., "El Lenguaje Unificado de Modelado", <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>