

USABILIDAD EN ENTORNOS MDA: PROPUESTA Y ESTUDIO EXPERIMENTAL¹

Silvia Abrahão¹, Emilio Insfran¹ y Jean Vanderdonckt²

1: Departamento de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

Camino de Vera, s/n, 46022

e-mail: {sabrahao, einsfran}@dsic.upv.es

2: Belgian Lab. of Computer-Human Interaction (BCHI)

Université Catholique de Louvain (UCL), School of Management (IAG)

Information Systems Unit (ISYS)

Place des Doyens, 1 – B-1348 Louvain-la-Neuve, Belgium

vanderdonckt@isys.ucl.ac.be, <http://www.isys.ucl.ac.be/bchi>

Palabras clave: MDA, Usabilidad, Evaluación, Interfaz de Usuario, Experimentación

Resumen. *Este artículo presenta un marco conceptual que incorpora la usabilidad en procesos de desarrollo de software basados en MDA (Model-Driven Architecture). En particular, se describe qué técnicas de evaluación pueden ser usadas y en qué momento. Para ilustrar la aplicación del marco se presenta un estudio experimental con un grupo de usuarios reales que evalúa la usabilidad de una aplicación generada automáticamente por una herramienta basada en MDA. Como resultado, se describe cómo los problemas de usabilidad identificados en el modelo de código (CM) pueden ser corregidos de forma iterativa en los modelos independientes plataforma (PIM) o en las reglas de transformación de código (PSM) dando soporte al concepto de “usabilidad obtenida por construcción”.*

1. INTRODUCCIÓN

La mayoría de organizaciones de desarrollo del software reconocen la relevancia de la usabilidad en el éxito de los productos software [8]. Sin embargo, este atributo de calidad no es debidamente tratado en las prácticas actuales de la Ingeniería del Software (IS). Por este motivo, se han propuesto algunos trabajos que abordan el problema de la integración de técnicas y actividades de usabilidad en procesos de desarrollo software [7] [11].

El desarrollo de software dirigido por modelos (DSDM) supone un nuevo enfoque de desarrollo basado en la separación de la funcionalidad esencial del sistema y la

¹ Este trabajo está subvencionado por el proyecto DESTINO con ref. TIN2004-03534 y desarrollado en el contexto de un proyecto I+D+I con CARE Technologies S.A.

implementación de dicha funcionalidad usando plataformas de implementación específicas. En particular, la propuesta *Model-Driven Architecture* (MDA) [15] promueve el uso de modelos durante el proceso de desarrollo y permite que éstos puedan ser transformados hasta la obtención del código fuente del software final. A pesar del crecimiento de la propuesta MDA, poco se ha considerado respecto a la evaluación de usabilidad en estos entornos y no hay propuestas sistemáticas que evalúan la usabilidad de las aplicaciones software obtenidas como parte de un proceso de transformaciones sucesivas.

En este artículo se presenta un marco conceptual genérico que incorpora la usabilidad en los procesos de desarrollo dirigidos por modelos. En particular, se describe qué técnicas de evaluación pueden ser usadas y en qué momento (modelo) del proceso. Las técnicas de evaluación son las utilizadas exitosamente en el ámbito de los procesos de desarrollo centrados en el usuario del campo de la Interacción Persona Ordenador (IPO). El marco propuesto es instanciado a un entorno de desarrollo MDA específico con el objetivo de mostrar: (i) cómo las actividades de evaluación de usabilidad pueden ser integradas con las demás actividades de un proceso de desarrollo dirigido por modelos y (2) cómo los problemas de usabilidad (PU) detectados como resultado de las actividades de evaluación pueden ser corregidos en diferentes modelos del proceso teniendo en cuenta las ventajas inherentes al paradigma MDA. El experimento realizado demuestra que las interfaces de usuario generadas automáticamente satisfacen un cierto nivel de usabilidad y que este nivel puede ser alcanzado por construcción, en oposición a la usabilidad de las interfaces obtenida por métodos artesanales tradicionalmente utilizados.

Este artículo está organizado como sigue. La sección 2 presenta una revisión de los métodos y ambientes de desarrollo de software dirigidos por modelos que han sido propuestos en los campos de la IS y de la IPO. La sección 3 presenta una propuesta de incorporación de la usabilidad en entornos MDA. La sección 4 presenta un estudio experimental realizado en el marco de la propuesta. La sección 5 presenta el análisis e interpretación de los resultados obtenidos. La sección 6 discute los problemas de usabilidad detectados y su impacto en los modelos PIM y PSM. Finalmente, la sección 7 presenta las conclusiones y trabajos futuros.

2. TRABAJOS RELACIONADOS

En los últimos años se han propuesto en el ámbito de la IS varios métodos y ambientes de desarrollo basados en modelos. Éstos utilizan normalmente un modelo de clases UML para capturar la estructura del sistema y un modelo de proceso para fijar la funcionalidad que el sistema debería proporcionar. Asimismo, existe una serie de herramientas CASE de carácter comercial o de propósito general (Together, UMT, AndroMDA, Rational Rose, Poseidon, etc.) con la finalidad de proporcionar algún tipo de automatización y gestionar estos modelos. La lista completa de estas herramientas puede encontrarse en [17].

Por otro lado, varios ambientes de desarrollo de interfaz de usuario (IU) basado en modelos han sido propuestos en el ámbito de la IPO. En [6] se revisan varios ambientes basados en modelos y se distinguen dos generaciones de herramientas. La primera generación proporciona un ambiente *run-time* para los modelos de interfaz de usuario y algunos ejemplos son: Cousin [9] y Humanoid [21]. La segunda generación proporciona soporte para el

modelado abstracto del interfaz de usuario en un alto nivel de abstracción. Ejemplos de estos ambientes incluyen Fuse [14], Trident [4] y Mecano [19]. Varios ambientes de la segunda generación se basan en un modelo de dominio que aporta una descripción de las entidades de dominio, representadas en un modelo declarativo de datos como en Mecano o un modelo orientado a objetos como en Fuse. Además, Fuse y Trident proponen modelos de tareas para representar los requisitos de interacción del usuario, del cual los modelos abstractos del interfaz (o los modelos de diálogo) se derivan más adelante.

Con la iniciativa MDA surgen otras propuestas como la de Jespersen y Linvald [12] que genera aplicaciones específicas de dominio en la plataforma J2EE a partir de una descripción declarativa del sistema, y la metodología de Vanderdonckt [22] que se expresa en tres ejes: modelos y sus lenguajes de especificación, método y por último las herramientas que apoyan el método basado en los modelos subyacentes.

Hasta ahora, se han iniciado varios intentos para el desarrollo sistemático de IU basado en modelos. Sin embargo, no se ha alcanzado todavía un consenso y ningún método estándar ha emergido. Aunque varios trabajos han sido conducidos en los campos de la IS e IPO para ver cómo MDA puede ser aplicado, ninguno de ellos integra explícitamente la usabilidad en los procesos de desarrollo MDA. Además, no tenemos referencias de ninguna propuesta que realice algún tipo de evaluación de usabilidad de las IU generadas en un entorno MDA.

3. INCORPORACIÓN DE USABILIDAD EN ENTORNOS MDA

Un proceso de desarrollo que sigue el enfoque MDA básicamente transforma modelos independientes de detalles de implementación (modelos PIM-*Platform-Independent Model*) en otros que aportan los aspectos específicos de una plataforma concreta (modelos PSM-*Platform-Specific Model*), hasta llegar al código fuente (modelos CM- *Code Model*).

La usabilidad de un sistema software obtenido como resultado de ese proceso de transformación puede evaluarse en distintos momentos de un proceso MDA: (i) en el CIM evaluando los modelos de requisitos, modelos de tareas o modelos de dominio que representan los requisitos de interfaz o de las tareas de usuarios; (ii) en el PIM evaluando los modelos que representan el interfaz abstracto del usuario como los modelos de diálogo y de presentación; (iii) en el PSM evaluando los modelos de interfaz concreto (si los hay); y (iv) en el CM evaluando el interfaz de usuario final.

Las actividades de evaluación que pueden ser realizadas en un proceso MDA se agrupan básicamente en dos grandes grupos: evaluación por expertos y tests de usabilidad. El objetivo de la evaluación por expertos (evaluaciones heurísticas, inspecciones de conformidad, análisis de acciones, etc.) es descubrir problemas de usabilidad usando como instrumento guías y estándares. El objetivo de los test de usabilidad (pensar en voz alta, test en laboratorio, etc.) es incluir al usuario en el proceso de evaluación. Se trata de la técnica más habitual en los procesos de desarrollo IPO. La premisa de esta técnica es que no se puede asegurar cuán usable es un producto software, sin antes probarlo con usuarios representativos llevando a cabo las tareas para las cuales el sistema ha sido diseñado.

La Figura 1 presenta un marco conceptual que describe las correspondencias entre los modelos de un proceso MDA genérico y las actividades de evaluación de usabilidad.

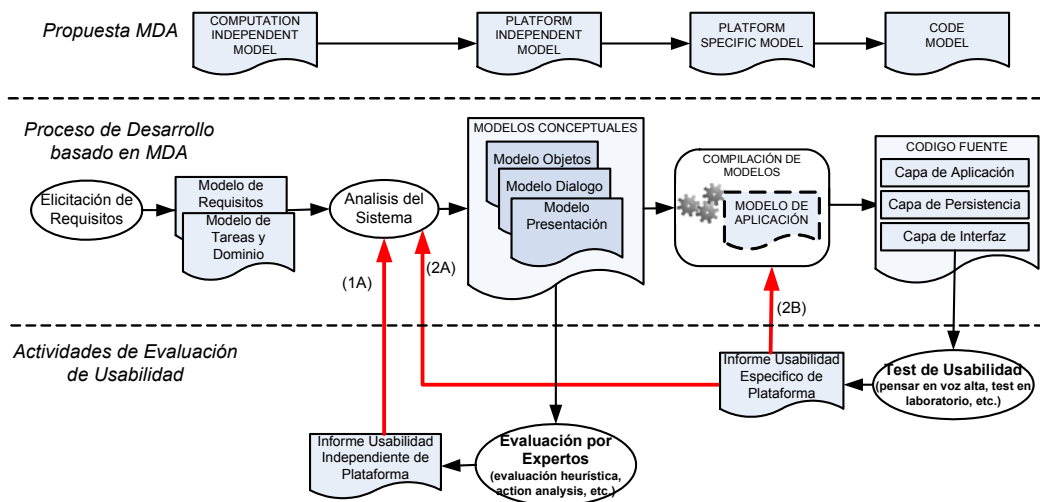


Figura 1. Correspondencias entre las actividades de un proceso MDA y actividades de evaluación de usabilidad

Habitualmente, la evaluación de la usabilidad se realiza sobre el CM. En un contexto MDA, la evaluación de usabilidad puede realizarse sobre los modelos abstractos del interfaz de usuario. Analizado el PIM² (modelos de diálogo, modelos de presentación, etc.) se obtiene como salida un *informe de usabilidad independiente de plataforma* que realimentará el análisis del sistema (1A). Por medio de las transformaciones entre modelos y la trazabilidad explícita entre éstos, los cambios realizados en el PIM son reflejados directamente en el CM, evitando de esta forma problemas de usabilidad en la aplicación a generar (CM).

Este es un proceso iterativo que se repite hasta que el PIM tenga el nivel de usabilidad requerido. La evaluación del PIM se puede realizar aplicando técnicas como la evaluación heurística o la inspección de estándares [13]. En [1] se describe un ejemplo de evaluación temprana de usabilidad usando un modelo de calidad.

Por otro lado, existen otros factores de usabilidad (eficiencia, satisfacción, etc.) que deben ser evaluados sobre una plataforma específica: sobre los componentes concretos del interfaz (PSM) o los componentes (widgets) que conforman la interfaz gráfica resultante (CM). En el CM, se puede aplicar técnicas que no requieren la participación del usuario (evaluación heurística, inspecciones, análisis de acción) o técnicas que incluyan la participación activa del usuario como los tests de usabilidad [13]. En [2] se describe un ejemplo de evaluación de usabilidad sobre el CM usando la técnica análisis de acción.

Finalmente, como resultado de la actividad de evaluación se produce un *informe de usabilidad específico de plataforma*. Sin embargo, en lugar de sugerir cambios en la capa de interfaz (CM), como es lo habitual en otros trabajos, en esta propuesta los cambios sugeridos se realizan en los modelos independientes de plataforma (2A). Otro beneficio sin precedentes en la literatura MDA, es la oportunidad de realizar una retroalimentación a las estrategias de transformación entre los modelos (2B).

² Aunque la evaluación también podría realizarse sobre los CIM (modelos de requisitos, de tareas y de dominio) en este trabajo nos centramos en las transformaciones entre los modelos PIM-PSM-CM.

3.1 Instanciación del Marco Conceptual a un entorno de desarrollo MDA

En esta sección describimos la aplicación del marco conceptual propuesto al entorno de desarrollo OlivaNova Model Execution (ONME) [5]. ONME sigue las directrices de MDA para la obtención de una aplicación software final realizado por transformaciones sucesivas de modelo. El desarrollo de aplicaciones en ONME se basa en la definición de un *modelo conceptual* (PIM) que se estructura en cuatro vistas complementarias: (1) Modelo de Objetos que describe las propiedades estáticas del sistema; (2) Modelo Dinámico que describe los aspectos relacionados con el control, las vidas válidas y la interacción entre objetos; (3) Modelo Funcional que describe la semántica asociada a los cambios de estado de los objetos; y finalmente, el (4) Modelo de Presentación, que permite la especificación abstracta de los requisitos de interfaz de usuario.

Finalmente, un proceso de *compilación de modelos* establece las reglas de transformación específicas para una plataforma tecnológica (PSM) del modelo conceptual (PIM) a la aplicación final (CM).

A fin de aplicar el marco conceptual, a continuación describimos el Modelo de Presentación, el principal modelo afectado por la usabilidad, que se especifica por medio de un lenguaje de patrones denominado Just-UI [16] estructurado en tres niveles:

- Nivel 1. Árbol de jerarquía de acciones: expresa cómo la funcionalidad será presentada jerárquicamente al usuario.
- Nivel 2. Unidades de interacción (UI): modela abstracciones de interfaz con la que el usuario interactuará. Existen cuatro unidades: *UI de servicio* que modela la presentación de un diálogo para que el usuario ejecute un servicio; *UI de instancia* que modela la presentación de los datos de una instancia; *UI de población* que permite mostrar un conjunto de instancias de una clase usando mecanismos de filtrado y ordenación para facilitar la selección y consulta de objetos; *UI de maestro/detalle* que modela unidades de interacción construidas a partir de UI más sencillas.
- Nivel 3. Patrones elementales: permiten restringir el comportamiento de las diferentes IU. Por ejemplo, la *selección definida* permite al analista enumerar los elementos válidos para un argumento de servicio y el *patrón de información complementaria* permite especificar información adicional que será mostrada al usuario.

4. DISEÑO DEL EXPERIMENTO

Un test de usabilidad ha sido realizado en diciembre de 2005 con el objetivo de evaluar la usabilidad de los IU generados por la herramienta ONME. Siguiendo el paradigma Goal-Question-Metric (GQM) [3] el objetivo del experimento ha sido:

Analizar el interfaz de usuario generado

Con el propósito de evaluar una aplicación en producción

Con respecto a su usabilidad

Desde el punto de vista de un grupo de usuarios reales de la aplicación

En el contexto de su entorno de trabajo simulado en un laboratorio de usabilidad

La aplicación a ser evaluada es el sistema de gestión de la empresa AdB, una compañía de abastecimiento de agua ubicada en la Comunidad Valenciana. Esta aplicación fue totalmente generada a partir de su modelo conceptual en la plataforma .NET.

4.1 Variables

La ISO 9241-11 [10] define la usabilidad como *el grado en que un producto puede ser utilizado por usuarios específicos para lograr objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado*. Por lo tanto, las variables dependientes del experimento son: la efectividad, la eficiencia y la satisfacción.

La efectividad es la completitud y precisión con la que los usuarios realizan ciertas tareas y se mide usando las métricas del *Common Industry Format* (CIF³):

- *Tasa de compleción no asistida*: porcentaje de subtareas realizadas con éxito sin la intervención del instructor.
- *Tasa de compleción asistida*: porcentaje de subtareas realizadas con éxito con la intervención del instructor.
- *Frecuencia de asistencia*: número de asistencias realizadas por el instructor al usuario en cada tarea.

La eficiencia relaciona el nivel de efectividad alcanzado con la cantidad de recursos empleada y se mide en términos de la tasa de compleción de tarea dividida por el tiempo de la tarea. Finalmente, la satisfacción de los usuarios en la interacción (impresión subjetiva) se mide con el cuestionario QUIS (*Questionnaire for User Interface Satisfaction*) [20].

4.2 Sujetos

El experimento ha involucrado ocho usuarios reales, todos ellos empleados de las empresas AdB y CARE Technologies. El nivel de experiencia de los usuarios en el dominio de la aplicación (funciones del negocio) y en la tecnología ONME se han considerado como variables independientes. La tabla 1 presenta la distribución de usuarios en cuatro grupos: noveles en el dominio y en la tecnología (grupo 1), noveles en el dominio y expertos en la tecnología (grupo 2), expertos en el dominio y noveles en la tecnología (grupo 3), y expertos en el dominio y en la tecnología (grupo 4). De esta forma, se verifica si el nivel de experiencia de los usuarios afecta a su rendimiento y/o satisfacción en el uso de la aplicación.

Tabla 1. Distribución de los sujetos

Usuarios (N=8)		Experiencia en el Dominio	
		Noveles	Expertos
Experiencia en la tecnología	Noveles	user7 user8	user5 user6
	Expertos	user3 user4	user1 user2

³ <http://zing.ncsl.nist.gov/iusr/documents/cifv1.1b.htm>

4.3 Objeto experimental

El objeto de la evaluación fue dos tareas representativas de la aplicación AdB seleccionadas a partir de un cuestionario realizado a los usuarios de la aplicación: *crear abonado* y *crear producto en stock*. La primera es una tarea compleja dependiente del dominio mientras que la segunda es una tarea medianamente compleja independiente del dominio de la aplicación. Ambas tareas se describen con la notación CTT (*ConcurTaskTrees*) de Paternò y otros [18]. El modelo de tareas para *crear abonado* se presenta en la Figura 2. Esta tarea incluye varias subtareas con un porcentaje de compleción para cada una.

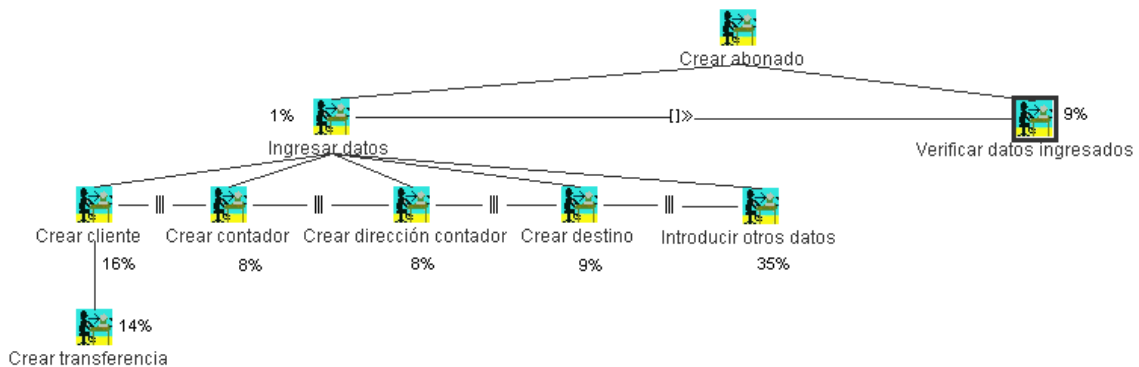


Figura 2. Modelo de tarea para la tarea crear abonado

La Figura 3 presenta el modelo de tareas para *crear producto en stock* con sus respectivas subtareas y porcentaje de compleción.

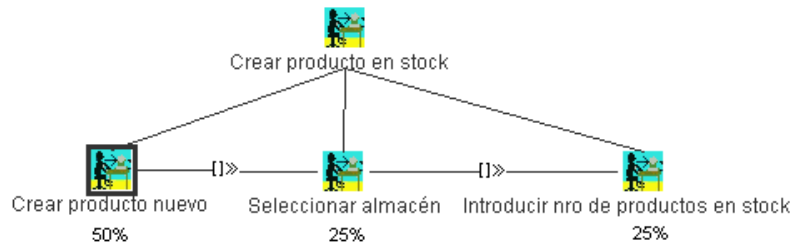


Figura 3. Modelo de tarea para la tarea crear producto en stock

4.4 Procedimiento e Instrumentación

Se ha instalado un laboratorio de usabilidad en CARE Technologies para conducir el experimento. La interacción del usuario con la aplicación AdB ha sido grabada en vídeo. Un instructor y un observador participaron en el estudio. Mientras que el primero interactuó con el usuario el segundo registró observaciones, ayudas, e incidencias ocurridas.

El procedimiento del test de usabilidad siguió una secuencia de eventos:

- Los usuarios completaron un cuestionario demográfico para confirmar sus datos personales y su actitud hacia la tecnología en general, el uso de ordenadores, el uso de A dB y su experiencia con la tecnología ONME, en una escala de 1 a 5.
- Los usuarios realizaron las dos tareas descritas en la sección 4.3. Para evitar un posible efecto *ceiling*, no hemos establecido límite de tiempo para las tareas.
- Una vez completada las tareas, los usuarios han rellenado un cuestionario de satisfacción (QUIS). Este cuestionario⁴ tiene un formato jerárquico con seis escalas para medir las reacciones de los usuarios y cuatro medidas de factores específicos del interfaz: *factores de la pantalla*, *terminología y retroalimentación del sistema*, *factores de aprendizaje y capacidades del sistema*. Cada uno de estos cuatro factores del interfaz se mide en una escala Likert de siete puntos. Además, se incluye como opción el ítem “no aplicable”. El QUIS también proporciona espacio adicional para que los usuarios hagan comentarios.

5 ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

Los datos demográficos revelaron que el 62% de los sujetos eran hombres, el 62% de ellos tenían entre 20 y 39 años, y el 25% de los sujetos tenían experiencia en el uso de la aplicación que han evaluado.

5.1 Eficiencia y Efectividad

La tabla 2 muestra que los usuarios llevaron un promedio de 14 minutos para terminar la primera tarea. Los datos también muestran que el 37,5% de los usuarios terminaron con éxito la tarea sin asistencia y que el 87,5% la terminaron con asistencia.

Tabla 2. Medidas de eficiencia y efectividad de la tarea 1

Sujeto#	Tasa Compleción Tarea no Asistida (%)	Tasa Compleción Tarea Asistida (%)	Tiempo Tarea ^a (min)	Tasa Compleción/ Tiempo Tarea	Asis- tencias
1	39%	8%	21,0	1,86%	1
2	31%	17%	10,0	3,10%	2
3	31%	69%	41,0	0,76%	5
4	100%	0%	17,0	5,88%	0
5	100%	0%	09,0	11,11%	0
6	47%	9%	16,0	2,94%	1
7	31%	8%	9,0	3,44%	1
8	100%	0%	20,0	5,00%	0
Media	64%	6%	14,57	4,76%	1,25
Error Std.	12,8%	2,4%	1,96	1,17	0,59
Desv. Std.	34,1%	6,4%	5,1	3,10	1,66
Mínimo	31%	0%	9,0	1,86%	0
Máximo	100%	17%	21,0	11,11%	5

^a Tiempo que el usuario ejecuta la tarea sin asistencia (hasta la primera vez que el usuario pide asistencia). Se presenta estadística descriptiva para los siete participantes que completaron la tarea.

⁴ <http://www.dsic.upv.es/~sabrahao/QUIS>

Las subtareas 4 y 6 son las que los usuarios han requerido más asistencia. El usuario 3 es el único usuario que no ha podido completar la tarea. En términos de eficiencia, los usuarios 5, 4 y 8 han tenido un rendimiento mucho mejor que los demás. Esto sugiere que la eficiencia es independiente de la experiencia de los usuarios ya que estos pertenecen a grupos distintos. La Figura 4 muestra la efectividad y la eficiencia de los usuarios por categoría. El gráfico revela que los usuarios expertos en el dominio y noveles en la tecnología ONME tuvieron un desempeño mejor que los demás usuarios. Este es un buen indicador de la facilidad de uso de ONME ya que los usuarios noveles han podido completar una tarea compleja sin asistencia.

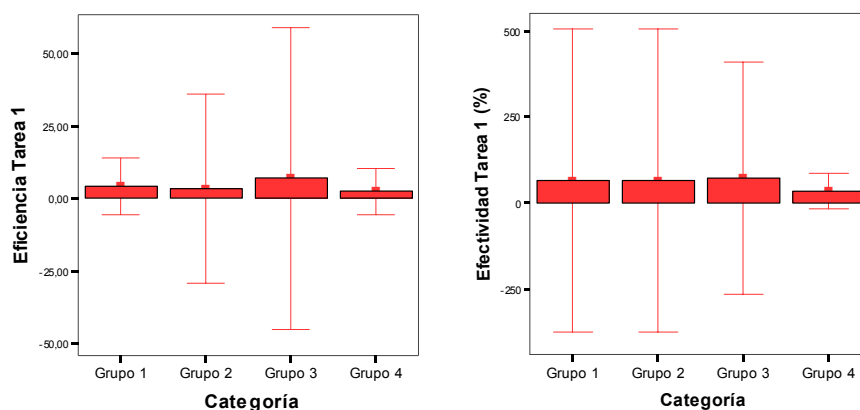


Figura 4. Medidas de eficiencia y efectividad de la tarea 1 por grupo de usuarios

La tabla 3 presenta los datos obtenidos para la segunda tarea. Los usuarios llevaron un promedio de 8 minutos para completar la tarea. En términos de efectividad, el 50% de los usuarios terminaron con éxito y sin asistencia la tarea, el 25% terminaron la tarea con asistencia y el 25% no han podido terminarla. Los usuarios 3 y 8 no han podido completarla.

Tabla 3. Medidas de eficiencia y efectividad de la tarea 2

Sujeto#	Tasa Compleción Tarea No Asistida (%)	Tasa Compleción Tarea Asistida (%)	Tiempo Tarea (min)	Tasa Compleción / Tiempo Tarea	Asistencias
1	50%	25%	7.0	7,14%	1
2	75%	25%	9.0	8,33%	1
3	0%	100%	12.0	0%	3
4	100%	0%	7.0	14,29%	0
5	100%	0%	4.0	25%	0
6	100%	0%	6.0	16,67%	0
7	100%	0%	15.0	6,67	0
8	0%	50%	0	0%	1
Media	87,5%	8,33%	8,0	13,01%	0,33
Error Std.	8,5	5,2%	1,5	2,91%	0,21
Desv. Std.	20,9	12,9%	3,7	7,14%	0,51
Mínimo	50%	0%	4,0	6,67%	0
Máximo	100%	25%	15,0	25%	1

Se proporciona estadística descriptiva solo para los seis participantes que completaron la tarea.

La Figura 5 presenta medidas de eficiencia y efectividad por grupo de usuarios. Los usuarios 5, 6 y 4 han tenido un rendimiento mucho mejor que los demás. Una vez más, los usuarios expertos en el dominio y noveles en la tecnología realizaron mejor la tarea que los demás usuarios.

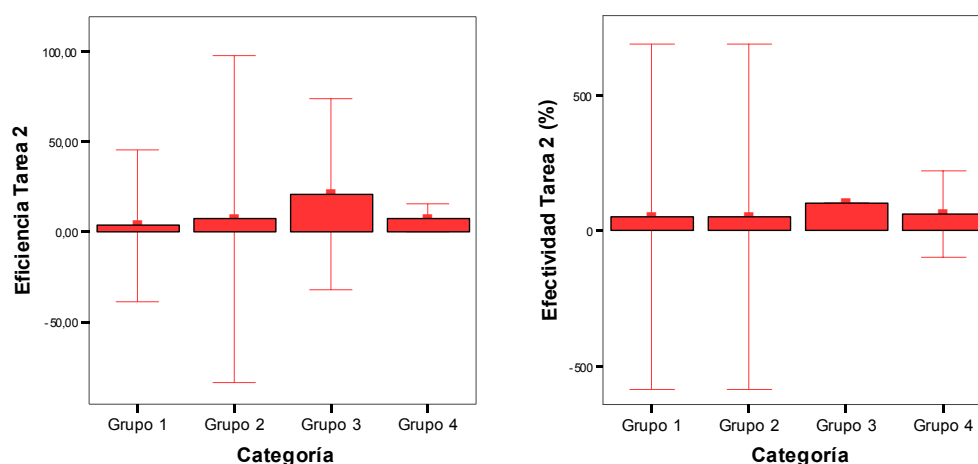


Figura 5. Medidas de eficiencia y efectividad de la tarea 2 por grupo de usuarios

5.2 Satisfacción

La Tabla 4 presenta los datos obtenidos para las medidas de satisfacción. Se muestra el valor medio de todas las respuestas por cada uno de los constructores del QUIS, basado en los valores obtenidos de la escala Likert, de 1 (peor) a 7 (mejor). Por tanto, el valor ‘neutro’ es el 4 y cuanto más alto el valor, mayor la satisfacción del usuario.

En general, las opiniones sobre la satisfacción de los usuarios con la aplicación fueron positivas ya que los valores medios superaron al valor neutro (4). El constructor *reacciones del usuario* ha quedado por debajo de la media, indicando que los usuarios han tenido dificultades para completar algunas tareas, como se ha comprobado en los vídeos.

Tabla 4. Estadística descriptiva para el QUIS

	Mínimo	Máximo	Media	Mediana	Desv. Std.
Reacciones del usuario	2,00	5,33	3,97	4,33	1,145
Factores de la pantalla	2,25	6,75	4,00	3,75	1,42
Terminología y retroalimentación del sistema	1,83	5,83	4,27	4,1	1,31
Factores de aprendizaje	1,83	6,40	4,57	4,80	1,39
Capacidad del sistema	4,60	6,0	5,35	5,40	0,57
Satisfacción	2,35	5,59	4,36	4,75	1,07

La Figura 6 muestra las medidas de satisfacción por categoría de usuario. Los usuarios noveles en el dominio y la tecnología juzgaron el sistema más satisfactorio que los otros tipos de usuario.

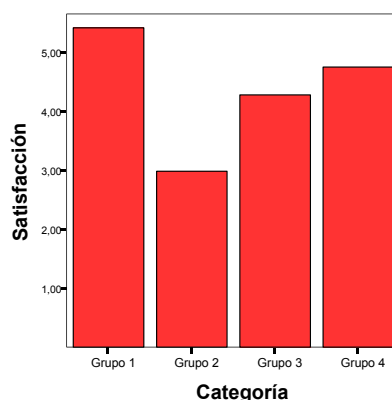


Figura 6. Satisfacción por grupo de usuarios

Sin embargo, para considerar estas medidas válidas, se realizó un análisis de la validez de construcción del QUIS por medio de una correlación inter-ítem del cuestionario. Como resultado, el constructor *capacidad del sistema* fue excluido lo que implicó que las correlaciones medias entre cada ítem quedaron entre 0,64 y 0,94; esto sugiere una buena correlación entre los ítems del cuestionario. También se ha realizado un análisis de la fiabilidad del QUIS con el alfa de Crombach y los resultados han sido: reacciones del usuario = 0,842; factores de la pantalla = 0,814; terminología y retroalimentación del sistema = 0,922 y factores de aprendizaje = 0,805. Estos análisis indican que el QUIS proporciona medidas válidas y fiables de la satisfacción del usuario.

6 PROBLEMAS DE USABILIDAD DETECTADOS Y SUS IMPLICACIONES EN EL PIM Y PSM

Debido a la transformación de modelos que implementa las relaciones entre el modelo conceptual (interfaz abstracto) y los componentes software del interfaz de usuario, los problemas de usabilidad detectados en el CM pueden ser corregidos en el PIM y PSM respectivos. Algunos de los problemas identificados en el experimento a través de los videos y registros de observación fueron: falta de etiquetas descriptivas, problemas de navegación (pasos cíclicos e innecesarios), mecanismos de búsqueda poco flexibles o incorrectos, ítems de menú y nombres de ventanas inconsistentes, falta de retroalimentación en selecciones de usuario y mensajes de error poco significativos. También verificamos que algunos de los problemas fueron confirmados por los sujetos con los comentarios hechos en el QUIS.

Una vez identificados los problemas, se debe determinar su origen (PIM o PSM) para elaborar un plan de recomendaciones de diseño y mejoras en el modelo conceptual de la aplicación o en las reglas de transformación de código de ONME.

A modo de ejemplo y considerando a las limitaciones de espacio, comentamos brevemente algunos problemas de usabilidad detectados y sus recomendaciones de cambio. El problema de los *mensajes poco significativos* afecta a la calidad de los mensajes de error. El siguiente mensaje de error apareció de forma frecuente durante la realización de la tarea 1: “La cardinalidad máxima para el abonado no se cumple. Como máximo una instancia de

abonado está permitida”. Aunque el mensaje informa sobre la forma de corregir el error, las indicaciones no son significativas para los usuarios finales. El origen de este problema está en el PIM (Modelo de Presentación) y corresponde a la propiedad “*mensaje de validación*” del *Patrón de Introducción* de una *UI de servicio* (véase la sección 3.1). Este patrón captura las características de los tipos de datos que el usuario introduce en el sistema e incluye la definición de máscaras de edición, valor por defecto, rango de valores, mensajes de ayuda y de error.

También hemos encontrado problemas de usabilidad que afectan a las reglas de transformación (PSM) de ONME. Un problema frecuente observado en la tarea 2 fue que los usuarios no podían encontrar la acción para *crear producto nuevo* (subtarea 1). Esto se debía a que el icono para ejecutar dicha acción quedaba oculto aunque la ventana se maximizara. Otros problemas identificados en este nivel fueron: la falta de conexión entre las funciones de la aplicación generada y la ayuda de referencia y la falta de actualización de los datos de un conjunto de visualización (rejilla) después de la ejecución de un servicio (por ejemplo, al *crear cliente*). Una posible solución sería refrescar los datos después de la ejecución de un servicio. Por otro lado, hemos verificado que algunos atributos de usabilidad son proporcionados automáticamente por el proceso de transformación. Por ejemplo, retroalimentación visual en menús y consistencia en la posición de los elementos de interfaz, entre otros.

7 CONCLUSIONES Y TRABAJOS FUTUROS

Se ha presentado un marco conceptual que incorporara la usabilidad en los procesos de desarrollo de software dirigido por modelos. Dicho marco ha sido instanciado en un entorno MDA específico con el objetivo de evaluar la usabilidad de las IU generadas automáticamente por la herramienta ONME. Para esto, se ha llevado a cabo un test de usabilidad con un grupo de usuarios reales que proporcionó una lista de problemas de usabilidad. Esta evaluación fue conducida sobre el interfaz de usuario final (CM) y proporcionó información valiosa para la mejora de los modelos PIM y PSM, así como el propio CM como consecuencia del proceso de transformación de código. Los resultados también han proporcionado evidencia empírica de la calidad intrínseca de las interfaces de usuario generadas por una herramienta MDA.

De esta forma, la evaluación de usabilidad incorporada a los procesos de desarrollo MDA facilita la obtención de aplicaciones software con un alto nivel de usabilidad debido a que las actividades de modelado-transformación-evaluación realizados de forma iterativa permite que la usabilidad sea obtenida por construcción y no ‘a mano’ como es habitual.

Como trabajo futuro se pretende explorar el marco propuesto en varias direcciones: utilizar otras técnicas de evaluación de usabilidad en distintos modelos (PIM-PSM-CM), estudiar la correlación entre los problemas identificados en los distintos niveles, determinar mecanismos para automatizar la corrección de ciertos problemas de usabilidad, e instanciar el marco propuesto a otras herramientas MDA.

REFERENCIAS

- [1] Abrahão S., O. Pastor, and Olsina L. (2005) A Quality Model for Early Usability Evaluation, INTERACT'05 International COST 294 Workshop on User Interface Quality Models, Rome, Italy.
- [2] Abrahão S., Iborra E. (2006) Usability Evaluation in the Context of Model-Driven Architecture: An Experimental Study, *Maturing Usability: Quality in Software, Interaction and Value*, Springer-Kluwer International Series in Human-Computer Interaction.
- [3] Basili V. y Weiss D. (1984). A Methodology for Collecting Valid Software Engineering Data, *IEEE Transactions on Software Engineering*, 10, 728-738.
- [4] Bodart F., Hennebert A., Leheureux J., Provot I., and Vanderdonckt J. (1994) A Model-Based Approach to Presentation: A Continuum from Task Analysis to Prototype. In Proceedings of DSV-IS'94, pp. 25-39, Bocca di Magra.
- [5] Care Technologies (2006) <http://www.care-t.com/products/index.html> Last access: May-2006
- [6] Da Silva, P. P. (2001) "User interface declarative models and development environments: A survey". Interactive Systems. Design, Specification, and Verification, 8th Int. Workshop, DSV-IS 2001, Glasgow, Scotland, Springer-Verlag Berlin.
- [7] Ferrer X. (2005) Marco de Integración de la Usabilidad en el proceso de Desarrollo software, Tesis doctoral, Universidad Politécnica de Madrid.
- [8] Hakiel S. (1997) Delivering Ease of Use, *Computing and Control Engineering Journal* Vol. 8, No. 2, pp. 81-87.
- [9] Hayes P., Szekely P. and Lerner R. (1985) Design Alternatives for User Interface Management Systems Based on Experience with COUSIN. Proc. of SIGCHI'85, Addison-Wesley, April 1985.
- [10] ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on usability (1998).
- [11] ISO/IEC International Standard: Information Technology. Software Life Cycle Processes. Amendment 1. ISO/IEC 12207:1995/ Amd. 1:2002.
- [12] J. W. Jaspersen and J. Linvald. Investigating user interface engineering in the model driven architecture. Interact 2003 Workshop on Software Engineering and HCI. IFIP Press, Sept. 2003.
- [13] Lewis J. and Rieman J. (1993) Task-Centered User Interface Design: A Practical Introduction (text at <ftp.cs.colorado.edu> in [/pub/cs/distrib/clewis/HCI-Design-Book](ftp://pub/cs/distrib/clewis/HCI-Design-Book)).
- [14] Lonczewski F., Schreiber S. (1996) The FUSE-System: an Integrated User Interface Design Environment. Computer-Aided Design of User Interfaces, Namur University Press, Belgium.
- [15] MDA: <http://www.omg.org/mda>. Last access: May-2006
- [16] Molina, P. J., S. Meliá, et al. (2002). JUST-UI: A User Interface Specification Model. Proceedings of Computer Aided Design of User Interfaces, CADUI'2002, Valenciennes, France.
- [17] OMG. <http://www.omg.org/mda/committed-products.htm>. Last access: 10-03-2006.
- [18] Paternò F., Mancini C et al. (1997) ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction, Chapman & Hall, Ltd.: 362-369.
- [19] Puerta A. (1996) The Mecano Project: Comprehensive and Integrated Support for Model-Based Interface Development. Computer-Aided Design of User Interfaces CADUI'96, pp. 19-36, Namur, Belgium, Namur University Press.
- [20] Shneiderman, B. (1993). Designing the User Interface. Strategies for Effective Human-Computer Interaction. Segunda edición, Addison Wesley, Reading, MA..
- [21] Szekely, P. (1990) Template-Based Mapping of Application Data to Interactive Displays. Proceedings of UIST'90, pp. 1-9. ACM Press, October 1990.
- [22] Vanderdonck, J. A MDA-Compliant Environment for Developing User Interfaces of Information Systems, 17th Int. Conference on Advanced Information Systems Engineering (CAiSE 2005), Porto, Portugal, June 2005, LNCS 3520, Springer 200, pp. 16-31 (keynote speech).