

# **N\_InterMod: Una Propuesta de notación de Diálogo enriquecida para el desarrollo ágil de aplicaciones interactivas.**

**Begoña Losada**

Dept. de Lenguajes y Sistemas  
Informáticos  
Facultad de Informática  
Universidad del País Vasco  
20001 San Sebastián  
[b.losada@ehu.es](mailto:b.losada@ehu.es)

**Maite Urretavizcaya**

Dept. de Lenguajes y Sistemas  
Informáticos  
Facultad de Informática  
Universidad del País Vasco  
20001 San Sebastián  
[maite.urretavizcaya@ehu.es](mailto:maite.urretavizcaya@ehu.es)

**Isabel Fdez de Castro**

Dept. de Lenguajes y Sistemas  
Informáticos  
Facultad de Informática  
Universidad del País Vasco  
20001 San Sebastián  
[isabel.fernandez@ehu.es](mailto:isabel.fernandez@ehu.es)

## **Resumen**

En el desarrollo de software interactivo con objetivos de usabilidad, un paso necesario es la descripción de las acciones del usuario y las respuestas del sistema; esto forma la base del modelo del diálogo. Desde el área de la ingeniería del software y de la ingeniería de la interfaz se han propuesto para ello varias notaciones. Sin embargo, la mejora de los procesos de desarrollo de software así como la de los propios modelos, nos ha llevado a plantear una nueva notación. Incluido en la metodología de desarrollo de software ágil InterMod , y como una parte de ella, en este artículo nos centramos en una propuesta para la notación del diálogo que incluye, además del propósito de usabilidad, objetivos de comunicación entre todos los agentes implicados en el desarrollo de la aplicación.

## **1. Introducción**

Este trabajo se enmarca en dos áreas complementarias. Por una parte, el desarrollo de software basado en modelos y, por otra, las metodologías ágiles para la gestión de proyectos. El desarrollo de software basado en modelos [7], permite diseñar el sistema con un nivel mayor de abstracción, eliminando detalles innecesarios en un punto dado del análisis. Las metodologías basadas en modelos [6][5], son estrategias habituales en el desarrollo de software. Actualmente, además para la gestión de proyectos, se usan metodologías ágiles, como XP y Scrum, cuyo objetivo es la obtención de resultados parciales probados por el usuario. Larman [3] define la agilidad como la respuesta rápida y flexible al cambio.

InterMod [4], es una metodología para desarrollo de aplicaciones interactivas basada en modelos, que propone la cooperación entre los diferentes actores implicados y una gestión ágil de productos validados en etapas tempranas. Para ello, un aspecto fundamental es el modelo de diálogo enriquecido que centra el presente artículo.

Habitualmente el modelo de diálogo concierne al orden y estructura de las acciones del usuario junto con las respuestas del sistema. Sin embargo, tal y como indica Dix [2], el modelo del diálogo es la estructura en la que se basa la semántica de la aplicación (su lógica de negocio). Por ello, el modelo de diálogo es la conexión necesaria entre los modelos de desarrollo de la interfaz y los de la implementación.

En el siguiente apartado haremos un repaso a las principales notaciones empleadas en la descripción del diálogo Hombre-Máquina (H-M), destacando sus ventajas e inconvenientes. En el apartado 3 presentamos nuestra propuesta de notación para el modelo de diálogo, N\_InterMod. Y finalmente, añadiremos unas conclusiones y comentaremos las líneas futuras de trabajo.

## **2. Modelo del diálogo**

En contraste con la mayoría de las conversaciones humanas, el diálogo H-M es más exigente, con una estructura más rígida y restringida que hace necesario la utilización de una notación formal para su modelado. Las justificaciones para el uso del modelo de diálogo Dix [2] son: (1) facilitar el análisis del diálogo, (2) separar los elementos de la interfaz y de la lógica de negocio, (3) permitir al diseñador analizar la estructura propuesta, o tal vez, le permite usar una herramienta de prototipado para ejecutar el

diálogo, y (4) Favorecer la comunicación entre los miembros de un equipo, ya que les permite debatir sobre aspectos de diseño y, llegados a un consenso, transmitir el modelo de diálogo.

Las notaciones formales usadas para la descripción del diálogo H-M pueden agruparse en: *esquemática*, mediante diagramas que permiten al diseñador observar de un vistazo la estructura del diálogo, y *textual*, mediante descripciones textuales que detallan los eventos sistema-usuario. En esta revisión valoraremos 7 características que nos permitan establecer el potencial de las notaciones, su legibilidad y su independencia sobre una filosofía concreta de codificación de la funcionalidad. Así, valoramos la posibilidad de describir diálogos *concurrentes* (1), *accesos generales* (2), *legibilidad de la notación* (3), *agrupación de las descripciones* (4), *orden y semántica* (5) de los eventos; *independencia de la notación* (6) con respecto al lenguaje de implementación, y *notación como medio de comunicación* (7) entre los agentes.

Por la diversidad de modelos de diálogo esquemáticos, presentamos una comparativa de las características mencionadas en la tabla 1. Las características destacadas las marcamos con una X y sombreamos aquellas que no se dan. En sí mismas, ninguna de las propuestas es una buena herramienta de comunicación (7), aunque en su mayoría por su simplicidad, tienen una legibilidad adecuada (3).

	1	2	3	4	5	6	7
STN			X		X	X	
STN de Harel	X		X		X	X	
Grafos	X		X	X	X	X	
Petri Nets	X	X			X	X	
Flujos tradicionales			X		X		
Árboles			X	X	X	X	
Diagramas UML	D	D			X		

Tabla 1. Comparativa de notaciones esquemáticas

Dentro de las notaciones textuales distinguimos dos grupos. Por una parte, las que derivan de una estructura formal (gramáticas, reglas de producción o las álgebras de proceso) y, por otra parte, las de aspectos narrativos (casos de uso y escenarios). El primer grupo destaca porque permite la descripción formal y precisa de los

acontecimientos. Sin embargo, estos aspectos van en detrimento de la legibilidad y por supuesto, dificultan su uso como herramienta de comunicación. El segundo grupo por el contrario son herramientas de comunicación útiles y directas [8]. Pero, es poco formal y puede contener descripciones ambiguas.

### 3. La notación para el diálogo N-Intermod

El modelo de diálogo no sólo debe recoger los requisitos funcionales de la aplicación, si no que debe favorecer el desarrollo de otros modelos en etapas tempranas. Para conseguir alcanzar los objetivos marcados es necesario paliar los inconvenientes de los formalismos vistos. Para ello proponemos N-InterMod, una notación de modelo de diálogo *esquemático* con aspectos *textuales* que facilita la comunicación. El formalismo escogido es el de los **grafos** visualizados como **árboles** pero permitiendo *accesos generales*. Además con el objetivo de agilizar los desarrollos, incluimos en la fase temprana del modelado del diálogo, elementos que simulan la semántica del programa y el aspecto final del producto. Proponemos obtener prototipos cercanos al resultado final y convertirlos en herramientas para verificar los requerimientos, funcionales y no funcionales, entre el usuario y los desarrolladores.

Así, la notación enriquecida del modelo de diálogo N-InterMod incluye: (a) El *modelo de tareas* con los requisitos funcionales, (b) el *modelo de navegación y comportamiento* que definen el flujo del devenir de la aplicación y las interacciones correctas e incorrectas, y (c) el *modelo de presentación* que describe elementos y modos de presentación como ventanas, secciones, colores o tipos de botones.

#### 3.1. Requisitos funcionales: Diseño de las acciones del usuario

La obtención del modelo de tareas es la fase previa de nuestro modelo de diálogo. Se establecen los pasos y orden en que el usuario deberá interactuar con el sistema con el fin de alcanzar los objetivos propuestos.

El analista, tras escuchar al usuario, diseña el nuevo escenario de la futura aplicación siguiendo

una notación en **árbol**. Los nodos de la jerarquía (Figura 1) representan tareas del usuario y se caracterizan por: una numeración ordenada con respecto a los padres, hijos y hermanos, un nombre descriptivo sobre dicha tarea y unos símbolos que permiten especificar el orden de ejecución de la tarea con respecto a sus hermanas, y el tipo de la tarea (opcional, repetitiva, etc.). Además, cada tarea tiene asociada una descripción narrativa que facilita y favorece todo el proceso de comunicación con el usuario. La figura 1 muestra una tarea de la jerarquía.

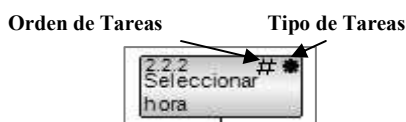


Figura 1. Ejemplo de una tarea del usuario

La especificación del orden de las tareas y su tipo, se establece mediante símbolos (el primero) en la esquina derecha superior del nodo que describe la tarea:

- Orden entre las tareas: secuencial (sin símbolo, orden por defecto) se realizan según un recorrido en profundidad, elección (o) sólo se puede realizar una de las tareas hermanas, indiferente (#) se pueden ejecutar en cualquier orden y concurrente (|||).
- Tipos de tareas de usuario: Unitaria (sin ningún símbolo-opción por defecto) se ejecuta una sola vez, opcional (?) se ejecuta 0/1 veces, y repetitiva (\*) se repite un número de veces definido en un intervalo.

A partir de esta primera versión funcional (parcial) del software que se quiere desarrollar, se puede obtener automáticamente un sencillo prototipo basado en menús y botones que facilitará las tareas de validación inicial. Tras la validación por el usuario de esta primera fase de desarrollo, se debe diseñar el modelo de diálogo incorporando las intervenciones y comunicación del sistema con el usuario.

### 3.2. Navegación y comportamiento

El diseño de la navegación usuario-sistema consiste en modelar las sucesivas intervenciones del usuario y del sistema a través del nuevo escenario, la aplicación en desarrollo. El analista conjuntamente con el desarrollador crea el modelo de diálogo a partir del modelo de tareas ya

verificado en la fase previa de análisis. Aunque las tareas del sistema aparecen en modelos de diálogo conocidos UML [1], CTT [5], en nuestro caso sólo se incorporan intervenciones/actuaciones del sistema que producen una comunicación explícita del sistema. Nuestro modelo de diálogo es un **grafo** creado a partir del modelo de tareas del ejemplo anterior. Las acciones del usuario y las del sistema se distinguen porque las del usuario están numeradas mientras que las del sistema no y en su lugar aparece la imagen de un óvalo inscrito en un rectángulo. Por ejemplo, en la Figura 2, podemos observar un modelo de diálogo parcial donde la tarea 2.3.1 tiene como respuestas asociadas tres posibles tareas del sistema. Estas tareas implican variaciones de la navegación y corresponden al modelo de comportamiento. Las variaciones se expresan con la etiqueta "Ir a..." asociada a la tarea del sistema (representa implícitamente un grafo).

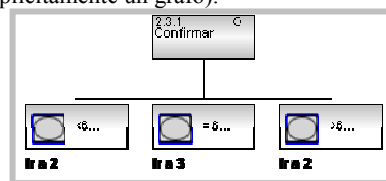


Figura 2. Ejemplo parcial del modelo de diálogo.

En este ejemplo se puede observar que en dos casos la tarea del sistema obliga al usuario a seguir con la tarea 2 (Ir a 2, probablemente por no estar completa/correcta del todo). Además, a cada respuesta del Sistema se le asocia un mensaje que informa al usuario sobre el comportamiento erróneo/desviado. En esta primera propuesta de modelo de diálogo, las tareas del sistema serán siempre tareas en nodos hojas de la jerarquía, y de tipo elección (sólo se ejecutará una de las acciones del sistema).

### 3.3. Enriquecimiento del Modelo de diálogo

Una vez diseñado el modelo de diálogo, o durante su diseño, se pueden ir estableciendo algunos aspectos sencillos de la apariencia de la interfaz de usuario, describiendo el modelo de presentación. Para ello, enriquezcamos los nodos del modelo de diálogo con aspectos de visualización. En esta primera propuesta se ha optado por incluir tres aspectos sobre la visualización del grupo de tareas (Figura 3): (A)

en una nueva ventana (trazo sencillo), (B) como tareas desplegadas bajo la tarea madre (doble trazo), o bien, (C) como tareas en una nueva sección de la misma ventana (trazo grueso). Se ha optado por una visión horizontal de la interfaz y se establecen las nuevas secciones siempre a la derecha de la actual. En una nueva ventana el menú que aparece está siempre en la primera sección. Cada nodo marca la visualización de sus descendientes en la interfaz, según las tres opciones señaladas.

Nueva ventana	Desplegadas	Nueva sección
2 Completar Tutorias	2.2 Selección día&hora	2.2.2 Seleccionar hora

Figura 3. Aspecto del modelo de presentación

Una vez que se ha completado el modelo de diálogo enriquecido para la fase actual del desarrollo de software establecido, se puede generar automáticamente un prototipo para su evaluación. En éste quedará constancia no sólo de la funcionalidad asociada al objetivo alcanzable por el usuario, sino los aspectos de interacción hombre-máquina, los de presentación en la interfaz y aquellos de navegación habitual u otros establecidos por el sistema.

#### 4. Conclusión y Líneas Futuras

El modelo de diálogo enriquecido propuesto se caracteriza por la sencillez y legibilidad de los diagramas en árbol. Permite incluir acciones concurrentes y accesos generales, establecer el orden y tipo de las acciones, así como el agrupamiento semántico.

La propuesta que presentamos favorece la producción ágil de trozos completos, correctos y validados (en todos los aspectos mencionados) de la aplicación final. Es decir, facilita la creación de herramientas para la comunicación entre los desarrolladores de la interfaz y de la funcionalidad, así como entre ellos y los usuarios finales. Este contacto a varios niveles es clave en las metodologías ágiles. En particular, la validación se realiza mediante la configuración y obtención de prototipos tempranos parciales del producto final. Estos prototipos simulan aspectos de presentación y funcionalidad.

En estos momentos estamos trabajando en una nueva versión de la notación y de las herramientas

que permitirán dividir las funcionalidades a desarrollar en subobjetivos enlazados pero con desarrollos y prototipos autónomos. De este modo facilitamos aún más el logro del objetivo de las metodologías ágiles; esto es, la producción ágil de trozos completos del resultado final. La validación de las propuestas actuales y futuras a corto plazo, se realizan con proyectos software docentes en la asignatura Interacción Persona Computador (FISS- UPV/EHU).

#### Agradecimientos

Este trabajo está financiado parcialmente por el Ministerio de educación a través del proyecto número TIN2009-14380, la Diputación de Gipuzcoa con el proyecto n. 157/2009 y el Gobierno Vasco con el proyecto n. IT421-10.

#### 5. Referencias

- [1] Booch, G., Rumbaugh, J., & Jacobson, I. (1999): The unified modelling language user guide. Reading, MA: Addison-Wesley
- [2] Dix., A, Finlay, J., Abowd, G., Beale, R. (2004): Human-Computer Interaction. PrenticeHall,.
- [3] Larman, C. (2007): Agile & Iterative Development. Addison-Wesley, Massachusetts
- [4] Losada, B., Urretavizcaya, M., Fernández-Castro, I. (2009): "The InterMod Methodology: An Interface Engineering Process linked with Software Engineering Stages", in New Trends on H-C Interaction, Springer.
- [5] Paternò, F. (2000): Model-Based Design and Evaluation of Interactive Applications. Springer, Great Britain.
- [6] Penichet, V., Lozano, M. D., Gallud, J. A., Tesoriero, R. User Interface Analysis for Groupware Applications in the TOUCHÉ Process Model. International Journal of Advances in Engineering Software, 2009.
- [7] Puerta, A. & Eisenstein, J (1999): Towards a General Computational Framework for Model-Based Interface Development Systems. Proc. ACM Conf. Intelligent User Interfaces, pp. 171-178,.
- [8] Rosson, M.B., J.M. Carroll. (2002): Usability Engineering. Morgan Kaufmann Publishers.