Representación abstracta de interfaces de usuario DGAUI

Javier Rodeiro Iglesias Susana Gómez Carnero

Escuela Superior de Ingeniería Informática de la Universidad de Vigo
Campus As Lagoas S/N, Ourense
{jrodeiro,susanagomez}@uvigo.es

Sumario

Las interfaces de usuario tienen un papel crucial en el éxito de una aplicación, a la vez que su desarrollo conlleva un importante coste tanto desde el punto de vista económico como temporal, con estos antecedentes es inevitable pensar que su optimización para conseguir una aceptabilidad máxima y un diseño efectivo sería de gran interés. Para que una interfaz se considere aceptable debe ser agradable a la vista del usuario, hacer lo que tiene que hacer y tener un uso fácil por parte del usuario. En este paper se presenta un modelo de representación de la interfaz que permita establecer validaciones sobre las tres premisas de aceptabilidad de interfaces de modo semiautomático, minimizando los costes de desarrollo y evaluación existentes actualmente. La representación que se propone en el modelo, considera que la interfaz de usuario visual no es una estructura continua sino que está compuesta por elementos finitos discretos, plantea la interfaz como una composición de elementos individuales denominados componentes, de estos componentes se indicará su descripción, las relaciones jerárquicas que se establecen entre ellos y la descripción de los cambios que provoca la realización de un evento sobre un componente dado.

Palabras clave

Diseño interfaces, usabilidad, modelos, representación abstracta

1. INTRODUCCIÓN

Dada la importancia de las interfaces de usuario en el éxito o fracaso de una aplicación y el tiempo y dinero que se invierten en su desarrollo, recordemos que Myers [Myers92] a través de una encuesta a desarrolladores lo cifra en un 48% y Garthner Group [Gartner94] dice que el 70% del esfuerzo de desarrollo de aplicaciones interactivas se dedica a la interfaz, es inevitable pensar que sería interesante poder optimizar las características de las mismas para conseguir una aceptabilidad máxima a través de un diseño más efectivo y consiguiendo de este modo un mejor aprovechamiento de los recursos.

La aceptabilidad de una interfaz de usuario está basada principalmente en tres cuestiones:

1. ¿Es la interfaz agradable a la vista del usuario? En este caso se habla de aceptabilidad de la interfaz basada en la estética de la misma. No se establecen criterios de utilización de la interfaz, si la aplicación subyacente de la interfaz es correcta o carga de memoria del usuario a la hora de utilizarla. Lo fundamental es que el usuario se sienta en un entorno agradable y amigable en cuanto a lo que ve, y que esta interfaz no le sea de ninguna forma repulsiva.

2. ¿Hace la interfaz lo que tiene que hacer? Se determina aquí si la interfaz es útil para el propósito con el cual fue creada, tanto en términos de que haga aquello que el usuario le indica que haga como en términos de que el usuario sea capaz de hacer aquello que desea en la interfaz. Este criterio esta relacionado con las tareas que el usuario tiene que realizar sobre la interfaz.

3. ¿Es fácil su uso por parte del usuario? Aquí se introduce el término de usabilidad [Shackel86] [Eason88] [ISO92] [ISO93] [Nielsen93] [IBM93] de la interfaz. Si un usuario encuentra demasiado complejo el manejo de una interfaz, por muy agradable que esta sea y por muy eficiente que sea el proceso que desarrolla, no la usará. La evaluación del cumplimiento de estos criterios es muy compleja. Esta complejidad viene dada porque el estudio de los mismos suele ser en la mayoría de los casos mediante inspección subjetiva, principalmente basada en opiniones de expertos o en encuestas sobre usuarios [Molich90] [Preece94] [Nielsen94] [Wharton94].

Existe una gran cantidad de técnicas de representación de interfaces de usuario, como indica Vanderdonckt [Vanderdonckt94] muchas de estas técnicas son exportadas de otras áreas como puede ser el diseño gráfico. Otras de estas representaciones vienen de formalismo matemáticos como el álgebra. Debemos examinar estas representaciones para comprobar si pueden ser aplicadas a interfaces complejas y establecer de una forma objetiva el cumplimiento de los criterios identificados anteriormente.

Por ello se plantea como objetivo del trabajo encontrar un modelo de representación de la interfaz que permita establecer validaciones sobre las tres premisas de aceptabilidad de interfaces de modo semiautomático, minimizando los costes de desarrollo y evaluación

existentes actualmente. Para ello en la sección 2 se presenta una alternativa que permite la representación de los aspectos visuales y de comportamiento de la interfaz. En la sección 3 se presenta un ejemplo de aplicación de la notación y en la sección 4 se extraen conclusiones y se plantean los trabajos futuros.

2. REPRESENTACIÓN DGAUI

Se ha realizado una revisión de la literatura existente sobre modelos de representación de interfaces de usuario, centrándose en aquellos modelos que proponían una representación visual de la interfaz y que permitían la definición del comportamiento de la misma [Gomez05], pero considerando los problemas que se han encontrado en dichas representaciones se presenta una solución alternativa.

La representación que se propone, DGAUI, considera que la interfaz de usuario visual no es una estructura continua sino que está compuesta por elementos finitos discretos, plantea la interfaz como una composición de elementos individuales denominados componentes, dichos componentes siguen una estructura topológica jerárquica, de forma que pueden estar contenidos unos dentro de otros.

Para la definición de la IU el diseñador gráfico dispone de un sistema de representación en el que puede:

- Realizar la definición de los componentes de la IU, posibilitando la creación de jerarquías de composición y la definición de nuevos componentes.
- Indicar la composición de componentes individuales para formar la interfaz de usuario concreta con la que el usuario interacciona.
- Representar el diálogo entre componentes dentro de la interfaz de usuario, indicando los eventos a los que atiende cada componente y de qué manera responde la IU a los mismos

Se entenderá que un **componente** de la interfaz de usuario es un elemento individual de la misma que es perceptible por el usuario a nivel de apariencia visual, responde a una acción del usuario, o bien sirve de elemento agrupador de otros componentes.

Cada componente tiene unas características propias que son independientes de la interfaz. Se consideran inherentes al componente, además de poseer unas propiedades relativas a su posicionamiento y comportamiento que se derivan de su participación o papel dentro de la IU en la que están incluidos.

Para una mejor interpretación y procesamiento de la representación de la interfaz parece adecuado estructurarla y las características de XML se adecuan perfectamente.

Por ello se plantea una estructuración de la representación basada en XML mediante la creación de una DTD que permita un parseado sencillo de la estructura de la representación.

Atendiendo a la semántica de la representación, en donde se encuentra por un lado la representación inicial de la interfaz, y después de un procesamiento, la estructura de los estados de la misma y sus transiciones, parece adecuado el separar asimismo estas dos representaciones. La primera (a la que llamaremos DGAUI-DEF) consiste en una definición detallada de cada uno de los elementos que conforman una interfaz y que es adecuado tener separados para permitir reutilización de la representación. La segunda (a la que llamaremos DGAUI-INT) es dependiente de la primera pues se calcula de la misma, aunque también sería factible construirla directamente si el concepto de diseño de la interfaz se orienta a estados y no a componentes como es el caso de DGAUI.

De esta forma, además, se permite una separación base entre presentación de la interfaz y funcionalidad de la misma. La presentación de la interfaz se haya en la definición de la representación mientras que la funcionalidad se haya en los estados y el tránsito entre estados a través de las acciones de los usuarios sobre los componentes y de los eventos del sistema.

Por tanto, la descripción inicial de la interfaz de usuario, DGAUI_DEF, se realizará mediante un documento XML que contiene la siguiente información:

- Composición de los componentes contenidos dentro de los contenedores.
- Información sobre la representación visual de los componentes.
- Información sobre los eventos y respuestas sobre componentes ante la ocurrencia de ese evento

<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT DGAUIDEF (Composicion, Descripcion, Dialogo)>

<!ELEMENT Composicion (Componente+)>

<!ELEMENT Descripcion (Grafico* | Texto* | Enumeracion*)*>

<!ELEMENT Dialogo (ItemDialogo*)>

Cada **componente** contenedor está formado por:

<!ELEMENT Componente (Alineado?, Equiespaciado?, Subcomponentes, InfI?, InfO?)>

Indica el nombre del componente, opciones de detalle sobre la alineación de los componentes incluidos, la lista de componentes incluidos y si el componente permite información de entrada o salida.

- *InfI*: El componente realiza introducción de información por parte del usuario. Se indica la información a través de un Id. Es opcional.

<!ELEMENT InfI (#PCDATA)>

- *InfO*: El componente muestra información desde la aplicación. Se identifica la información a través de un Id. Es opcional.

<!ELEMENT InfO (#PCDATA)>

La **descripción** de cada componente de la interfaz indica que puede ser un gráfico, un texto o un gráfico definido por enumeración.

<!ELEMENT Descripcion (Grafico* | Texto* | Enumeracion*)*>

*Componente gráfico***:** viene definido por su nombre, una primitiva, el estilo, ancho y color de línea, su color de relleno, su posición y su tamaño.

<!ELEMENT Grafico ((Rectangulo | Linea | Circulo | Elipse | Poligono)?, EstiloLinea?, AnchoLinea?, ColorLinea?, ColorRelleno?, Posicion?, Tamano?, Datos?)>

Tanto los componentes gráficos, como los componentes tipo texto y los gráficos por enumeración tienen un conjunto de propiedades comunes que tendrán valor verdadero o falso. Estas propiedades son:

- Activo: indica si el componente puede responder a eventos.

- Visible: indica si el componente es visible para el usuario en el dispositivo de visualización.

- InfI: permite activar la función de entrada de datos asociada al componente. Es opcional.

- InfO: permite activar la función de salida de datos asociada al componente. Es opcional.

<!ATTLIST Grafico

Nombre CDATA #REQUIRED

Visible (t | f) #REQUIRED

Activo (t | f) #REQUIRED

InfI (t | f) #IMPLIED

InfO (t | f) #IMPLIED >

La primitiva que define el componente puede ser cualquiera de las primitivas de dibujo usadas generalmente en los paquetes gráficos.

La *posición* de un componente determina su posición dentro de su componente contenedor o bien dentro del dispositivo. Dicha posición puede ser fija o relativa indicada mediante restricciones topológicas que se indican en el campo OpCRel, en cualquier caso esta posición viene dada por una coordenada que indica su posición superior izquierda.

<!ELEMENT Posicion (Fija | Relativa)>

<!ELEMENT Fija (Coordenada)>

<!ELEMENT Relativa (OpCRel?, Coordenada)>

<!ELEMENT OpCRel (Centrado | (Justificado, Justificado?))>

- *Componente de texto*: viene definido por el texto que representa, un tipo, tamaño y color de fuente; además también posee las características de posición y tamaño así como las características de visible, activo, InfI e InfO ya explicadas en componentes gráficos.

<!ELEMENT Texto (Txt, Fuente, TamanoFuente, ColorFuente, EstiloFuente, Posicion, Tamano)>

- *Componentes gráficos por enumeración*: incluye la ruta al fichero que contiene la imagen correspondiente al gráfico; además incluye valores para indicar el nombre, posición, tamaño, su visibilidad, y actividad y posibilidad de entrada y salida de datos por parte del usuario ya descrito en componentes gráficos.

<!ELEMENT Enumeracion (Fichero, Posicion, Tamano)>

El **diálogo** sobre los componentes de la interfaz está formado por el componente sobre el que se produce el diálogo, el evento que lo provoca y los cambios efectuados en las propiedades de otros componentes (La respuesta al evento).

<!ELEMENT Dialogo (ItemDialogo*)>

<!ELEMENT ItemDialogo (Precondiciones?, Respuesta)>

<!ATTLIST ItemDialogo

Elemento CDATA #IMPLIED

Evento CDATA #REQUIRED >

<!ELEMENT Precondiciones (Precondicion+)>

<!ELEMENT Precondicion (#PCDATA)>

<!ATTLIST Precondicion

Visible (t | f) #IMPLIED

Activo (t | f) #IMPLIED

InfI (t | f) #IMPLIED

InfO (t | f) #IMPLIED >

<!ELEMENT Respuesta (Cambio)+>

<!ELEMENT Cambio (#PCDATA)>

<!ATTLIST Cambio

Visible (t | f) #IMPLIED

Activo (t | f) #IMPLIED

InfI (t | f) #IMPLIED

InfO (t | f) #IMPLIED

ProcID CDATA #IMPLIED >

La representación de los componentes de la interfaz es fija para la representación de la interfaz. Los aspectos dinámicos producidos por la interacción del usuario con la interfaz están reflejados en la representación del diálogo de los componentes.

En la representación del diálogo se asocia a cada componente el evento al cual responde junto con los cambios que este provoca en la interfaz.

3. Experimentación

Como ejemplo simple se presenta la modelización de un botón Next cuya apariencia sería:

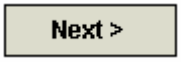

Figura 1: Botón Next

La acción que se podría realizar sobre él sería la selección del mismo, obteniendo la siguiente apariencia:

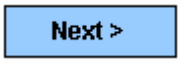

Figura 2: Botón Next seleccionado

Se presenta a continuación la representación XML del botón.

En la composición se identifican dos componentes independiente Button_Off y Button_On.

```
<DGAUIDEF>
 <Composicion>
  <Componente Nombre="Button_Off">
   <Subcomponentes>
    <Cont>Button_off_text</Cont>
   </Subcomponentes>
  </Componente>
  <Componente Nombre="Button_On">
   <Subcomponentes>
    <Cont>Button_on_text</Cont>
   </Subcomponentes>
  </Componente>
 </Composicion>
```

La descripción presenta los componentes contenedores independientes (Button_Off y Button_On) como dos primitivas rectángulo y los subcomponentes (Button_off_text y Button_on_text) como componentes de tipo Texto.

El fichero de especificación completo de este ejemplo se puede obtener en [Gomez05]

4. Conclusiones y Trabajos futuros

Se ha mostrado una nueva representación que hace especial énfasis en la parte visual de la interfaz, relacionándose muy fuertemente esta parte visual con el comportamiento de la interfaz frente a las acciones del usuario.

Se ha planteado un ejemplo concreto del uso de la notación para un componente simple, pudiéndose deducir de los resultados obtenidos su aplicabilidad sobre múltiples componentes. Siendo factible obtener de forma automática el comportamiento de los componentes.

Como trabajo futuro se plantea la presentación de la notación que permita representar la funcionalidad de las interfaces (DGAUI_INT) así como la construcción de una interfaz compleja como podría ser la de un procesador de texto.

Una vez modelada la interfaz se plantea la extracción del conjunto de tareas de usuario desde la especificación como base para la evaluación de la interfaz.

5. AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto TIN2005-08863-C03-02 y por el proyecto 05VI-C02.

6. REFERENCIAS

[Eason88] Eason, K. Information Technology and Organizational Change. Taylor and Francis, London. 1988.

[Gartner94] Gartner Group Annual Symposium on the Future of Information Technology, Cannes 7-10 november 1994.

[Gomez05] Gómez Carnero, S, Rodeiro Iglesias, J. Aplicación de los sistemas de representación para la sistematización de la validación de interfaces de usuario. *Technical Report TR-LSI-GIG-05-2. Computer Science Department. University of Vigo*. 2005. `www.ei.uvigo.es/~susanagomez/hci.html`

[IBM93] IBM, IBM Dictionary of Computing. McGraw-Hill 1993.

[ISO92] ISO. Software product evaluation quality characteristics and guidelines for their use.

[ISO93] ISO. Ergonomics Requirements for Office Work with Visual Displays Terminals: Guidance and Usability.

[Molich90] Molich R. y Nielsen J. Heuristic evaluation of user interfaces in *Proceedings of ACM CHI 1990*. Seattle, WA, Abril 1990, Pág. 249-256, 1990.

[Myers92] Myers B. A. y Rosson M. B. Survey on user interface programming, in *CHI'92 Conference Proceedings on Human Factors in Computing Systems* (Bauersfeld P.,Bennett J. y LYNCH G. eds.), pág. 195-202. ACM Press, Nueva York, NY, 1992.

[Nielsen93] Nielsen, J. Usability Engineering. Academic Press, London. 1993.

[Nielsen94] Nielsen J. y Mack R. L. eds. Usability Inspection Methods. John Wiley and Sons, New York, NY, 1994.

[Preece94] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. And Carey Tom. Human-Computer Interaction. Addison-Wesley Publishing, Reading, Mass. 1994.

[Rodeiro01] Rodeiro Iglesias, J. Representación y análisis de la componente visual de la interfaz de usuarios. PhD Thesis. Universidad de Vigo. Septiembre 2001.

[Shackel86] Shackel, B. Ergonomics in designing for usability. In *M. D. Harrison and A. Monk, editors. People and Computers: Designing for Usability*. Cambridge University Press, 1986.

[Vanderdonckt94] Jean Vanderdonckt, Xavier Grillo. Visual techniques for traditional and multimedia layout. June 1994.

[Wharton94] Wharton C. ET AL. The cognitive walkthrough method: a practitioner's guide en Usability Inspection Methods (NIELSEN J. y MACK R. L. eds.). John Wiley & Sons, New York, NY, Pág. 105-140, 1994.