

# Revisión de Lenguajes Declarativos para la Descripción de Interfaces de Usuario Independientes del Dispositivo

Nuria Hurtado Rodríguez  
Departamento de Lenguajes y  
Sistemas Informáticos  
Universidad de Cádiz  
Escuela Superior de Ingeniería  
C/ Chile s/n  
956 015296  
nuria.hurtado@uca.es

J. Mariano González Romano  
Departamento de Lenguajes y  
Sistemas Informáticos  
Universidad de Sevilla  
ETS de Ingeniería Informática  
Avda. Reina Mercedes s/n  
954552768  
mariano@lsi.us.es

Jesús Torres Valderrama  
Departamento de Lenguajes y  
Sistemas Informáticos  
Universidad de Sevilla  
ETS de Ingeniería Informática  
Avda. Reina Mercedes s/n  
954552769  
jtorres@lsi.us.es

## RESUMEN

En este artículo se presenta una revisión de los principales lenguajes basados en XML para la descripción de interfaces de usuario independientes del dispositivo, realizando una comparativa entre ellos y destacando su relación con el proceso de modelado de interfaces de usuario basado en modelos declarativos y sus fases o niveles de abstracción.

## Palabras Clave

Lenguajes declarativos de descripción interfaces de usuario, desarrollo de interfaces de usuario basado en modelos, interfaces independientes del dispositivo, XML.

## 1. INTRODUCCIÓN

El principal objetivo del desarrollo de interfaces de usuario basado en modelos es la construcción de interfaces mediante descripciones de alto nivel de los distintos aspectos de la interfaz: estructura y comportamiento, de modo que a partir de dichos modelos declarativos se pueda generar automáticamente la interfaz de usuario final [4].

Basándose en esta idea, se han construido diversos entornos de desarrollo y herramientas que evitan la necesidad de utilizar un lenguaje de programación concreto. En su lugar, el diseñador utiliza notaciones de mayor nivel de abstracción para especificar estos modelos o descripciones declarativas de la interfaz.

Estos lenguajes de modelado, describen la interfaz de usuario en diferentes niveles de abstracción según la fase del proceso de modelado en la cuál pueden ser utilizados. Algunos lenguajes pueden servir para especificar todos los modelos declarativos que componen el modelado de la interfaz, es decir pueden ser usados durante el proceso de desarrollo completo. Sin embargo, otros

Se concede el permiso para la reproducción digital o impreso total o parcial de este trabajo sin contraprestación económica únicamente para la utilización personal o en clase. En ningún caso se podrán hacer o distribuir copias de para su explotación comercial. Todas las copias deben de llevar esta nota y la información completa de la primera página. Para cualquier otro uso, publicación, publicación en servidores, o listas de distribución de esta información necesitara de un permiso específico y/o el pago correspondiente.

Interacción 2004, 3-7 mayo, 2004, Lleida (España).

sólo pueden ser utilizados en determinados niveles [10].

Al mismo tiempo, este enfoque basado en modelos declarativos proporciona un marco idóneo para la definición de interfaces de usuario independientes de contexto de uso, ya que permite utilizar descripciones de alto nivel de abstracción que pueden modelar aspectos independientes del contexto, o de las restricciones impuestas por un dispositivo o plataforma específicos.

Se presenta a continuación un esquema del componente principal del proceso de modelado de interfaces de usuario basado en modelos declarativos. Posteriormente se realiza un repaso de las principales características de cada uno de los lenguajes revisados destacando su relación con dicho proceso. Finalmente se presenta una comparativa de los principales aspectos contemplados.

## 2. MODELADO DE LA INTERFAZ

En la figura 1 se muestra el esquema general del modelo de interfaz. Este modelo declarativo está estructurado generalmente en tres niveles de abstracción principales. En el primer nivel podemos situar los modelos de tareas y dominio. El segundo nivel del modelo llamado Especificación de Interfaz de Usuario Abstracta representa la estructura y contenido de la interfaz en términos abstractos. Especifica de un modo abstracto la información que se mostrará en cada ventana y el diálogo para interactuar con dicha información. El tercer nivel, llamado Especificación de Interfaz de Usuario Concreta especifica el estilo para la representación de las unidades de información [12].

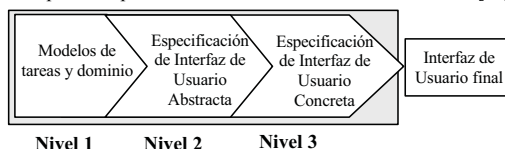


Figura 1.- Modelado de la interfaz

### 2.1 Notación para la descripción de los modelos

Para la especificación de los modelos, los entornos de desarrollo de Interfaces de Usuario basados en modelos (MB-UIDE)

incorporan determinadas notaciones descriptivas que van desde especificaciones algebraicas, representaciones gráficas, hasta la utilización de lenguajes de especificación propios de cada entorno.

Una de las mayores desventajas del desarrollo de Interfaces basadas en Modelos es la complejidad que presentan algunas de estas notaciones que a menudo resultan difíciles de aprender y utilizar [9].

Para evitar esto, los MB-UIDE proporcionan herramientas de modelado que sirven a los desarrolladores para construir dichos modelos. El principal objetivo de estas herramientas es ocultar a los desarrolladores la sintaxis de los lenguajes de modelado y proporcionarles una interfaz que les permita especificar adecuadamente el modelo de interfaz. Se ha desarrollado una amplia gama de herramientas de modelado, a menudo especializadas en diferentes niveles del modelo [13].

Podemos observar que no existe una notación estándar para la descripción de los diferentes modelos dentro de los MB\_UIDE desarrollados. Cada MB-UIDE utiliza una notación diferente, ya sea gráfica o textual, e incluso un mismo MB-UIDE utiliza diferentes notaciones o lenguajes de modelado para los diferentes modelos declarativos de que se compone el modelo de interfaz.

### 3. LENGUAJES DE DESCRIPCIÓN DE INTERFACES BASADOS EN XML

La definición de interfaces de usuario utilizando lenguajes declarativos proporciona varias ventajas, como la facilidad de aprendizaje y la posibilidad de definir la interfaz de forma independiente de la definición de la lógica de la aplicación y el contenido. De esta manera se permite que diferentes especialistas trabajen independientemente en el desarrollo de la aplicación [8].

Los lenguajes basados en XML se están perfilando como serios candidatos a soportar las especificaciones gracias a la versatilidad de mantenimiento, extensión y capacidades de refinamiento que proporcionan los documentos XML [5].

La tecnología XML, como estándar de representación común, permite la especificación del modelo de interfaz abstracto, la descripción de las características específicas de los diferentes dispositivos, así como la especificación del proceso de transformación de los objetos de interacción abstractos en objetos de interacción concretos [6].

#### 3.1.1 AAIML (*Alternate Abstract Interface Markup Language*)

AAIML es un lenguaje basado en XML en vías de desarrollo que se enmarca dentro de un proyecto más amplio para el desarrollo de un protocolo estándar para el acceso a interfaces alternativas. El objetivo de este proyecto es ampliar el mercado, haciendo más accesibles las interfaces y los dispositivos, por ejemplo a personas con discapacidades.

El hecho de que los dispositivos electrónicos estén fabricados por diferentes compañías, requiere la definición de un estándar que permita utilizar una consola "alternativa" para controlar los dispositivos específicos. La especificación del concepto de Consola Remota Universal (URC) es una parte esencial del desarrollo de la definición de dicho estándar.

El concepto de URC permite a las personas con o sin discapacidades controlar remotamente cualquier dispositivo electrónico desde su dispositivo personal de control remoto que puede encontrarse situado en cualquier lugar [15].

El lenguaje AAIML debe servir para especificar la definición de una interfaz de usuario abstracta para un determinado servicio o dispositivo. Dicha interfaz sería transmitida desde el dispositivo a la URC. Este lenguaje debería ser lo suficientemente abstracto, para permitir que un dispositivo URC particular pueda traducir la interfaz proporcionada a los mecanismos de E/S concretos del dispositivo adaptado a las necesidades del usuario [15].

#### 3.1.2 AUIML (*Abstract User Interface Markup Language*)

El lenguaje AUIML es un lenguaje basado en XML desarrollado por IBM y diseñado para permitir la definición de la semántica de la interacción con el usuario. Está centrado, por tanto, en la descripción de aspectos de interacción más que en aspectos de presentación.

Toda la información de la interacción se codifica una sola vez y se traduce utilizando una traducción dependiente del dispositivo final. Está diseñado para ser independiente de la plataforma, del lenguaje de programación final y de la tecnología de implementación [2].

Consta de dos principales conjuntos de elementos, los que se representan a través del modelo de datos, que definen la estructura de la información necesaria para soportar la interacción con el usuario, y los que se representan a través del modelo de presentación que definen el estilo de la presentación.

El lenguaje AUIML está todavía en desarrollo y no existen herramientas que soporten de forma completa su especificación, siendo ésta un documento de trabajo interno de IBM. AUIML estaría englobado dentro de los niveles 2 y 3 del proceso de modelado.

#### 3.1.3 UIML (*User Interface Markup Language*)

El lenguaje UIML es un sencillo lenguaje basado en XML que permite realizar una descripción declarativa de la interfaz de usuario de un modo independiente del dispositivo. Uno de los objetivos de UIML es reducir el tiempo que los desarrolladores invierten en describir interfaces para múltiples familias de dispositivos [1].

Para describir una interfaz de usuario en UIML se debe realizar, por un lado la definición de la interfaz genérica, y por otro un documento UIML que representa el estilo de presentación apropiado para el dispositivo en el cuál la interfaz de usuario se va a ejecutar. De este modo, una misma aplicación solamente necesitará un único documento UIML de especificación válido para cualquier dispositivo y un documento de estilo propio para cada dispositivo [14].

Podemos decir que UIML se podría utilizar en los niveles 2 y 3 del esquema de modelado presentado y parcialmente en el nivel 1. La definición de la interfaz se enmarcaría en los niveles 1 y 2 y la definición del estilo en el nivel 3. Estas tareas incluso podrían ser llevadas a cabo por equipos de desarrollo diferentes.

La flexibilidad para la selección de dispositivos finales es limitada, ya que aunque la parte correspondiente a la interfaz genérica puede mantenerse independientemente de que aumente el número de dispositivos finales, no sucede lo mismo con la parte que mapea a los dispositivos específicos, que crece cuando este número aumenta, aumentando también el coste de mantenimiento [3].

Permite la traducción automática al lenguaje utilizado por el dispositivo final. El proceso de traducción se realiza en el propio dispositivo o en el servidor de la interfaz dependiendo del dispositivo del que se trate [14].

### 3.1.4 XIML - (*eXtensible Interface Markup Language*)

El lenguaje XIML es un lenguaje de especificación basado en XML. Se propone como lenguaje de especificación común e infraestructura de desarrollo para profesionales de la interfaz de usuario en todos los ámbitos, diseñadores, ingenieros de software o expertos en usabilidad.

El lenguaje XIML se presenta como una propuesta de representación común para datos interactivos, ya que contempla los principales requisitos que debe cumplir un lenguaje de este tipo: Soporta funcionalidad a lo largo del ciclo de vida completo del desarrollo de interfaces de usuario, es capaz de relacionar los elementos abstractos y concretos de una interfaz, permite a los sistemas basados en conocimiento tratar los datos capturados [11].

XIML es una colección organizada de elementos de interfaz categorizados en uno o más componentes principales. En su primera versión los componentes son tareas, dominio, usuario, diálogo y presentación, extraídos del estudio de los modelos declarativos del enfoque basado en modelos. Los tres primeros componentes se podrían considerar abstractos y los dos últimos específicos [11].

Por tanto, podemos decir que XIML podría ser utilizado en los niveles 1, 2 y 3 del proceso de modelado de interfaces de usuario presentado. La independencia de la plataforma de uso se hace posible gracias a la estricta separación entre la definición de la interfaz y la traducción de la misma.

Existe un Foro de XIML en el cuál se incluyen representantes del mundo académico y de la industria, cuyo objetivo es promover la investigación, divulgación, adopción y estandarización de XIML. Hasta ahora, se han presentado resultados de la evaluación de sus requisitos y en este momento se encuentra en fase de expansión y divulgación.

### 3.1.5 XUL (*XML-based User-Interface Language*)

El lenguaje XUL es un lenguaje de descripción de interfaces basado en XML, específicamente diseñado para aplicaciones en red como navegadores, programas de correo. Está integrado dentro de la arquitectura de Mozilla para el desarrollo de Interfaces Web multiplataforma, dentro de la cuál se hace uso de tecnologías W3C ya existentes.

La arquitectura se basa en el uso de paquetes que pueden ser abordados desde una perspectiva abstracta o concreta. Los paquetes se componen de contenido, apariencia, comportamiento,

localización, plataforma. En cada uno de ellos se hace uso de diferentes tecnologías [4].

Su ejecución deberá realizarse bajo el entorno de Mozilla y en los Sistemas Operativos en los cuáles Mozilla se ejecute. Tiene la capacidad de separar la interfaz de la lógica de la aplicación, lo cuál facilita el mantenimiento de la interfaz sin necesidad de alterar la lógica de la aplicación.

Podríamos utilizar XUL para realizar la definición de la especificación de la interfaz concreta, nivel 3. Contempla parcialmente el nivel 2 a través del resto de las tecnologías de las que hace uso.

### 3.1.6 XForms

XForms es una propuesta del consorcio W3C para la especificación de formularios para la Web que puedan ser usados en una amplia variedad de plataformas. Su versión 1.0 ha llegado a ser recientemente una recomendación de W3C.

El desarrollo de XForms pretende cubrir las limitaciones de los formularios HTML tradicionales que no disponen de una separación entre el propósito y la presentación de un formulario. XForms se compone de secciones separadas que describen lo que el formulario hace y cómo se presenta [7].

La especificación de XForms está compuesta de dos módulos, el Modelo XForms y el soporte de la interfaz de usuario. El modelo XForms representa las diferentes partes del formulario desde el punto de vista de los datos genéricos. La interfaz de usuario XForms define la parte del formulario que representa los elementos de la presentación, proporcionando flexibilidad y control sobre la misma a través de las diferentes presentaciones que pueden estar relacionadas con un mismo modelo XForms, dependiendo de la plataforma final.

XForms se podría englobar fundamentalmente en el nivel 3 de abstracción (parte del nivel 2), ya que es un lenguaje orientado a la implementación de formularios Web en diferentes dispositivos pero no a la descripción de la interfaz en un nivel alto de abstracción.

## 3.2 Comparativa de Lenguajes

En la tabla 1 se muestra una comparativa de algunos aspectos relativos a los lenguajes declarativos estudiados: Niveles de abstracción del proceso de modelado en los que se puede situar cada lenguaje. Si es capaz o no de abarcar múltiples familias de dispositivos. Cuál es su ámbito de aplicación y si el tipo de interfaz generada es exclusivamente una interfaz gráfica de usuario (GUI) o puede contemplar otros tipos de interfaces (UI). Qué versiones de cada lenguaje se encuentran desarrolladas o especificadas, y si el lenguaje se encuentra o no en fase de mejora y expansión.

Con respecto a los modelos y niveles de abstracción del proceso de modelado de la interfaz de usuario, cabe destacar que el único lenguaje que contempla el ciclo de desarrollo completo de la interfaz de usuario es XIML, cubriendo todos los niveles de abstracción. Posiblemente, esto se derive de su evolución a partir de la investigación en el enfoque basado en modelos. El resto de los lenguajes presentados no cubre completamente el ciclo de desarrollo de un enfoque basado en modelos. Nos referimos a

AUIML, XUL y XForms que contemplan los niveles 2 y 3 exclusivamente. Su ámbito de aplicación es más limitado.

En el caso de UIML el objetivo es el desarrollo rápido de interfaces facilitando el prototipado de las mismas. Se centra en la definición de la interfaz abstracta en lo relativo a los aspectos independientes del dispositivo, y en su mapeado a la interfaz de usuario específica para cada dispositivo. Por tanto, su ámbito es más concreto y también más limitado en principio que el de XML, pero al igual que éste se presenta como un formato de intercambio universal que pretende ser un estándar y que orienta su desarrollo futuro hacia un lenguaje abstracto para la descripción de interfaces en un nivel alto de abstracción.

Algunas de estas propuestas pueden resultar complementarias, como es el caso de la generación de formularios de XForms a partir de una especificación UIML.

Lenguaje	Niveles	F.Disp	Ámbito	Desarrollo	Tipo
AAIML		Si	Global	Proyecto	UI
UIML	(1), 2, 3	Si	Disp. de red	1.0, 2.0, 3.0 En desarrollo	UI
XForms	(2), 3	Si	Web	1.0 En desarrollo	F.Web GUI
AUIML	(2), 3	(Si)	Global	En desarrollo	GUI
XUL	(2), 3	No	Aplic. de red	1.0 En desarrollo	GUI
XIML	1, 2, 3	Si	Global	1.0 En desarrollo	UI

Tabla 1 . Comparativa de lenguajes.

#### 4. CONCLUSIONES

En general, observamos una tendencia a la búsqueda de la estandarización, de un lenguaje de representación común para los datos interactivos y al mismo tiempo una evolución y continuo desarrollo de propuestas tanto desde el ámbito académico como desde el ámbito de la industria.

El estándar XML se presenta como una alternativa ideal para la especificación de Interfaces de usuario por su capacidad de extensión, su versatilidad y su relativa sencillez. Observamos que en general los lenguajes estudiados pueden ser utilizados para especificar adecuadamente algunos modelos declarativos como alternativa a otras notaciones complejas. También se comprueba que algunas de las metodologías y entornos de desarrollo de Interfaces de Usuario basados en modelos propuestos hasta el momento, tienden a contemplar el uso de alguno de estos lenguajes en alguna de las fases del proceso de modelado.

Los lenguajes basados en XML son especialmente adecuados para la especificación de interfaces independientes del dispositivo, plataforma y contexto de uso y ésta es, por tanto, una de las líneas de desarrollo presente y futuro de estas tecnologías.

#### 5. REFERENCIAS

[1] Abrams, M. Phanouriou, C. Batongbacal, A. Stephen M, Shuster W&J. UIML: An XML Language for Building Device-Independent User Interface Language. 8<sup>th</sup> International World Wide Web Conference, Toronto Conversion Centre, Toronto, Canada. May, 1999.

[2] Azevedo, P, Merrik, R. Roberts, D. OVID to AUIML- User Oriented Interface Modelling IBM UK. TUPIS 2000.

[3] Farooq, M. Abrams, M Simplifying Construction of Multi-Platform User Interfaces Using UIML. UIML Europe 2001 Conference. March, 2001.

[4] Lozano, M.D. Entorno Metodológico Orientado a Objetos para la Especificación y Desarrollo de Interfaces de Usuario. Tesis doctoral. Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia. Octubre, 2001.

[5] Molina, P.J. Especificación de Interfaz de Usuario: de los requisitos a la generación automática. Tesis doctoral. Departamento de Sistemas y Computación, Universidad Politécnica de Valencia. Marzo, 2003.

[6] Müller, A. Forbrig, P. Cap, C. Model-Based User Interface Design Using Markup Concepts. Proceedings of the 8<sup>th</sup> Workshop of Design, Specification and Verification of Interactive Systems. June, 2001.

[7] Nuno, D. Object Modelling for User Centered Development and User Interface Design: The Wisdom Approach. Ph D. Thesis. Department of Mathematics. University of Madeira. April, 2001.

[8] Phanouriou, C. UIML: A Device-Independent User Interface Markup Language, Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Applications September, 2000 Blacksburg, Virginia.

[9] Pinhero, P. User Interface Declarative Models and Development Enviroments: A survey. Department of Computer Science. University of Manchester. April, 2000.

[10] Pribeanu C. Vanderdonckt J., Limbourg Q. Task Modelling for Context Sensitive User Interfaces. Proceedings of 8<sup>th</sup> International Workshop on Design, Specification, Verification of Interactive Systems DSV-IS'2001, 49-68. Springer Verlag, Berlin, Germany. October, 2001.

[11] Puerta, A. Eisenstein. XML: A Universal Language for User Interfaces. White paper.

[12] Szekely, P. Retrospective and Challenges for Model-Based Interface Development. In: Vanderdonckt, J.: Proc. of 2nd Int. Workshop on Computer-Aided Design of User Interfaces CADUI'96 (Namur, June 5-7, 1996). Presses Universitaires de Namur. Namur (1996) xxi-xliv.

[13] Traetteberg H. Model Based User Interface. Ph. D. Thesis Department of Computer and Information Sciences. Faculty of Information Technology, Mathematics and Electrical Engineering. Norwegian University of Science and Technology. May, 2002.

[14] White Paper: The UIML Vision. Harmonia Inc. Blacksburg, Virginia USA. February, 2000.

[15] Zimmermann, G. Vanderheiden, G. Gilman A. Universal Remote Console Prototyping of an Emerging XML Based Alternate User Interface Access Standard. 11<sup>th</sup> International World Wide Web Conference. Honolulu, Hawaii. May 2002.