

# Estudio comparativo de diferentes propuestas dirigidas por modelos para la implementación de RIAs

Feliu Trias, Marcos López-Sanz, Esperanza Marcos

Kybele Research Group  
Rey Juan Carlos University  
Tulipán S/N, 28933, Móstoles, Madrid, Spain.  
{feliu.trias, marcos.lopez, esperanza.marcos}@urjc.es

**Abstract.** En la última década las aplicaciones Web han experimentado importantes cambios en cuanto a su arquitectura, las funcionalidades que ofrecen y las interfaces de usuario que presentan. Esta nueva forma de entender las aplicaciones Web recibe el nombre de RIA (*Rich Internet Application*), y queda patente en la denominada Web 2.0. Ante estas nuevas necesidades, las metodologías dirigidas por modelos para el desarrollo de aplicaciones Web se han visto obligadas a adaptar sus métodos para generar aplicaciones Web que ofrezcan estas nuevas prestaciones. Este trabajo presenta un estudio comparativo que analiza cómo diferentes métodos han sido adaptados al modelado y la generación automática de RIAs. Además de estudiar otros métodos que han surgido expresamente para la construcción de este tipo de aplicaciones.

**Palabras clave:** Rich Internet Applications, Desarrollo Dirigido por Modelos, Ingeniería Web.

## 1 Introducción

En la última década las aplicaciones Web han experimentado cambios significativos relacionados con la arquitectura, la distribución de la lógica de negocio entre la parte del cliente y la del servidor, la capacidad de almacenamiento de datos por parte del cliente, la comunicación que se establece entre ambas partes y las interfaces de usuario que proporcionan [1]. Esta evolución de la Web ha dado lugar a un nuevo paradigma de aplicaciones Web llamado RIA. Este nuevo tipo de aplicaciones ha tenido muy buena acogida entre los usuarios y cada vez son más demandadas ya que con ellas se vencen muchas de las limitaciones de las Webs tradicionales, además de ofrecer novedosas y atractivas prestaciones (p. ej. actualizar los contenidos dinámicamente o incluir contenido multimedia bajo demanda) [2].

En este nuevo escenario, la parte del cliente extiende sus funcionalidades en la arquitectura cliente-servidor ya que puede almacenar datos y ejecutar parte de la lógica de negocio. Este nuevo concepto de cliente recibe el nombre de *fat-client*

[3]. Algo muy diferente al cliente de las Webs tradicionales, conocido como *thin-client* [3], que se encarga únicamente de detectar eventos de usuario, permitir la entrada de datos y mostrar al usuario la información procesada por el servidor que alberga toda la lógica de negocio [4].

Esta ganancia de autonomía por parte del cliente permite que la comunicación con el servidor pueda realizarse de forma asíncrona [3], reduciendo de esta forma el tráfico de mensajes y permitiendo, por lo tanto, que las aplicaciones Web sean más eficientes.

En cuanto a las interfaces de usuario (IU), las RIAs proporcionan interfaces con un nivel de interactividad y usabilidad mucho mayor [5]. Una de sus principales características es la forma de actualizar su contenido. En la Web tradicional la mínima unidad de datos que se transmite es la página Web. Por este motivo, cuando se debe actualizar algún dato de la IU la página se genera de nuevo y se envía al cliente donde se carga [3]. En cambio, en las RIAs la mínima unidad transmitida es el dato. Por lo tanto, en la actualización de la IU no es necesario generar y cargar toda la página sino únicamente el dato que ha cambiado. Este nuevo concepto de IU recibe el nombre de *single-window paradigm* [2].

La generalización del uso de los principios de las RIAs ha propiciado un cambio en las metodologías y métodos definidos para el desarrollo de aplicaciones Web. En concreto, ha tenido una gran influencia en aquellas que siguen un enfoque dirigido por modelos. Durante esta última década han enfocado sus esfuerzos en adaptar sus procesos de modelado al desarrollo de RIAs. En este trabajo se presenta un estudio comparativo de las propuestas que plantean algunas metodologías para la adaptación de sus métodos al desarrollo de RIAs.

Este trabajo se estructura de la siguiente manera: la Sección 2 presenta los diferentes métodos y propuestas que serán objeto de nuestro estudio; la Sección 3 expone los criterios para poder desarrollar el estudio comparativo junto con los valores asignados a cada criterio de evaluación; la Sección 4 contiene los detalles de la comparativa y, por último, la Sección 5 recoge las principales conclusiones extraídas del estudio realizado.

## **2 Metodologías y propuestas para el modelado de RIAs**

El nuevo paradigma RIA ha convertido el modelado de las aplicaciones Web en un proceso mucho más complejo que el de la Web tradicional [1]. Por este motivo, en los últimos años las metodologías dirigidas por modelos han centrado sus esfuerzos en adaptar sus métodos al modelado y generación de RIAs.

Las principales aspectos que aportan las RIAs son: interfaces más complejas con un alto nivel de interactividad, clientes con capacidad de almacenar datos y de ejecutar parte de la lógica de negocio y la posibilidad de que la parte del cliente se comunique de manera asíncrona (además de síncrona) con el servidor [1].

Las propuestas que plantea cada metodología para su adaptación a las RIAs se centran en uno o varios de estos aspectos.

A continuación, se presenta brevemente cada metodología las diferentes propuestas que proponen. Las metodologías que serán materia de estudio son: OOH4RIA [6], OOHDM [7], RUX [8], WebML [4], UWE [9] y OOWS [10].

## 2.1 OOH4RIA

OOH4RIA es una propuesta para el modelado de RIAs que extiende el método OO-H [11]. Se centra en el aspecto de la IU proponiendo 2 nuevos modelos: el modelo de presentación y el de orquestación [6].

El modelo de presentación organiza y da estructura a la IU. Para ello, introduce los conceptos de *page*, *screenshot* y *widget*. Por otro lado, el modelo de orquestación modela el comportamiento existente entre los *widgets* que componen la IU y el comportamiento entre éstos y el sistema.

El proceso de construcción de la RIA empieza con la definición del modelo conceptual de datos. A partir de éste se obtiene de forma automática el modelo de navegación. A continuación, se genera el modelo de presentación mediante una transformación M2M (*Model to Model*), y seguidamente el modelo de orquestación. A partir del modelo de presentación y orquestación se genera el código que implementa la IU. La Fig. 1 presenta el proceso completo de modelado de OOH4RIA.

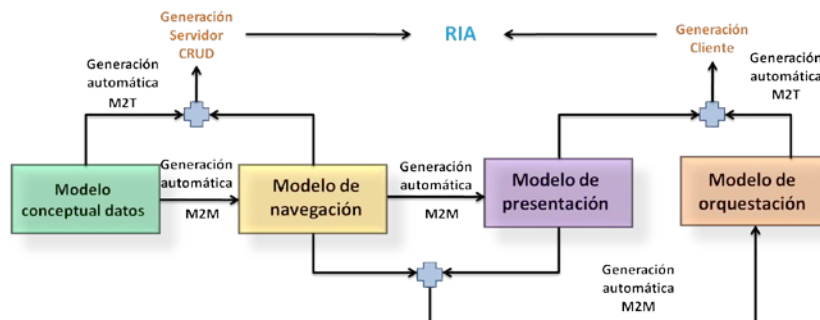


Fig. 1. Proceso de modelado de OOH4RIA

Otra propuesta planteada en OOH4RIA es la de adaptar las interfaces de usuario a los diferentes dispositivos de salida (PDA, móvil, PC, etc.). Para ello se propone la utilización del modelo de usuario (*user model*) [12]. Éste modelo junto al modelo de presentación genera de manera automática diferentes modelos de presentación específicos para un dispositivo. A partir de éste junto al modelo de navegación se genera el modelo de orquestación. La Fig. 2 muestra el proceso de modelado teniendo en cuenta la adaptación al dispositivo de salida.

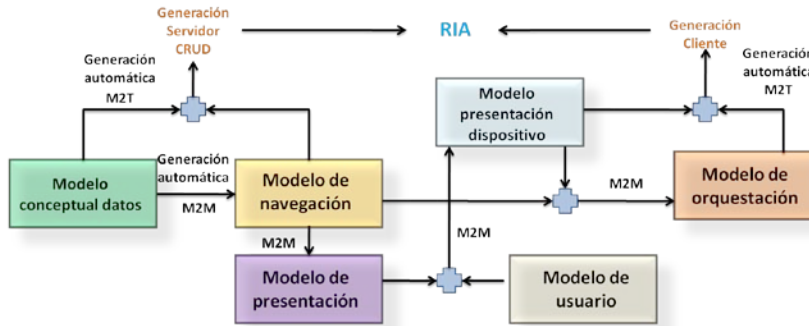


Fig. 2. Proceso de modelado de OOH4RIA con adaptación a dispositivo

## 2.2 OOHDM

El método OOHDM [13] también se extiende para dar soporte al desarrollo de las RIAs, y concretamente para el modelado e implementación de la IU. El proceso de modelado empieza con el modelo conceptual de datos, seguido del modelo de navegación. Para el modelado de la IU, se construye el modelo de ADVs (*Abstract Data View*)[14] que permite definir la estructura de la IU y los elementos que la conforman. Para modelar su comportamiento proponen la utilización del modelo ADV Chart, un diagrama de estados que permite modelar los cambios que experimenta la IU causados por los eventos detectados [7]. A partir del modelo de ADVs se obtiene la implementación de la interfaz en una cierta plataforma. La Fig. 3 muestra un modelo ADV.



Fig. 3. Proceso de modelado de OOHDM

## 2.3 RUX Method

Es un método específico para el modelado e implementación de IUs para RIAs. Se caracteriza por tener tres modelos para modelar la IU: modelo de interfaz abstracta, modelo de interfaz concreta y modelo de interfaz final [8].

El modelo de interfaz abstracta define la estructura de la interfaz mediante elementos llamados *views* que albergan diferentes componentes que constituirán la futura interfaz [8].

El modelo de interfaz concreta hereda mediante una transformación M2M la estructura de la interfaz definida en el modelo de interfaz abstracta y la adapta para

un dispositivo determinado (PC, PDA, móvil, etc.). Para ello, se detalla la forma, posición, color de cada uno de los componentes de la interfaz. Además, modela la comunicación entre los diferentes componentes de la IU, y las interacciones del usuario con ésta.

Por último, en el modelo de interfaz final se obtiene el código que implementa la IU mediante una transformación M2T (*Model to Text*).

Al ser un método específico para el modelado e implementación de IUs para RIAs necesita acoplarse con otro método. Métodos como WebML [4] y UWE [9] han utilizado RUX Method para el modelado y generación de sus IUs.

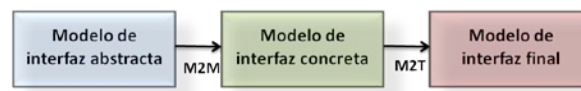


Fig. 4. Proceso de modelado de RUX Method

## 2.4 WebML

WebML es un método dirigido por modelos que también se ha adaptado al desarrollo de RIAs. Se centra en el aspecto que permite al cliente almacenar datos y ejecutar parte de la lógica de negocio [1] [4]. El proceso de modelado de WebML empieza con la definición del modelo de procesos de negocio implementado en BPMN (*Business Modelling Language*) [15]. A partir de éste se obtiene de manera automática el modelo conceptual de datos y el modelo de navegación [2]. La capacitación de almacenamiento de datos del cliente queda reflejada en el modelo conceptual de datos. Mediante etiquetas se especifica el lugar donde residirá el dato (cliente o servidor) y su nivel de persistencia (persistente o volátil) [16]. Por otro lado, en el modelo de navegación se realizará el reparto de la lógica de negocio. Las funcionalidades serán etiquetadas para definir en qué parte (cliente servidor) serán ejecutadas. El aspecto de la presentación no se modela mediante modelos, sino que se codifica de forma manual mediante código XHTML [17] páginas de estilo XSLT.

Otra propuesta de WebML es trabajar con RUX Method para la definición e implementación de la IU [16]. A partir de su modelo de navegación se genera el modelo de interfaz abstracta de RUX y empieza el modelado de la interfaz. La **¡Error! No se encuentra el origen de la referencia.** muestra el proceso de desarrollo de esta combinación.

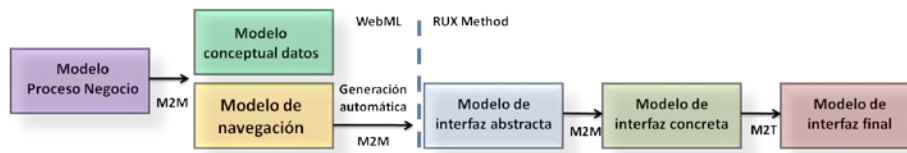


Fig. 5. Proceso de desarrollo de la combinación entre WebML y RUX Method

## 2.5 UWE

UWE es otro método dirigido por modelos para la construcción de aplicaciones Web. Su propuesta se enfoca en el comportamiento de la IU. El proceso de modelado empieza con la definición del modelo de requisitos y del modelo conceptual de datos. Seguidamente, y de forma automática, se genera el modelo de navegación [18]. A continuación, crea el modelo de presentación que define la estructura de la IU. Para la definición de su comportamiento propone la utilización de patrones, denominados *RIA patterns* [19]. Estos patrones se asocian a los elementos del modelo de presentación y con ellos se modela el comportamiento de los *widgets* que componen la IU. Estos patrones se asemejan a diagramas de estados de UML [20]. Otra propuesta que plantea para la definición y generación de la IU es la combinación con RUX Method. De este modo, a partir del modelo de presentación propio de UWE se obtiene el modelo de interfaz abstracta de RUX Method y empieza así el proceso de construcción de la IU [9]. La Fig. 6 muestra el proceso de modelado surgido de la combinación de UWE con RUX Method.

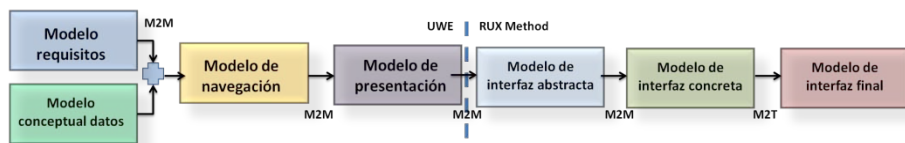


Fig. 6. Proceso de modelado de UWE combinado con RUX Method

## 2.6 OOWS

OOWS [21] es un método dirigido por modelos que propone la definición de un metamodelo que recoge los conceptos más destacados para modelar las IUs de las RIAs. Este metamodelo, llamado *RIA metamodel* [10], abarca tanto el aspecto estático como dinámico de este tipo de interfaces.

El proceso de modelado de OOWS empieza con la creación del modelo de usuario. A continuación, se crea el modelo de navegación. A los elementos de este modelo se les asocian AIPs (*Abstract Interaction Patterns*) para definir la interacción entre el usuario y el sistema [10]. Para modelar e implementar la IU relacionan mediante un modelo de *weaving* [22] el metamodelo de navegación y el metamodelo que define los AIPs con el *RIA metamodel* propuesto. Una vez tienen establecidas estas meta-relaciones generan el modelo de la interfaz para una tecnología específica mediante transformaciones M2M. La Fig. 7 muestra el aspecto dinámico del nuevo metamodelo.

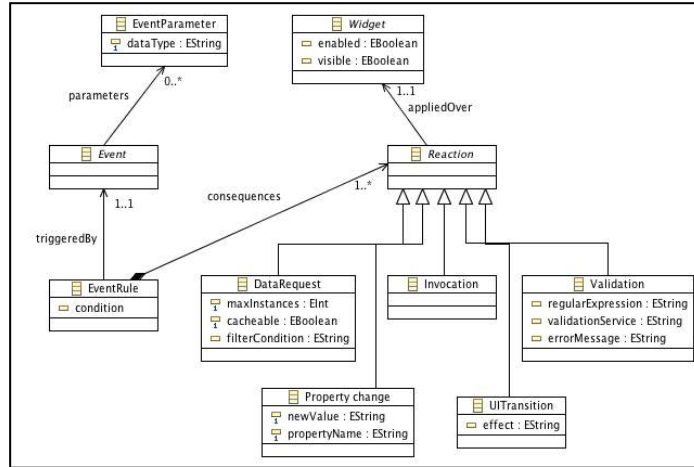


Fig. 7. Aspecto dinámico del metamodelo RIA

### 3 Criterios de comparación

Una vez se han presentado las propuestas más significativas para el desarrollo de RIAs, es necesario establecer un vocabulario común que permita realizar un análisis contrastado de todas ellas. Por ello, a continuación se exponen los diferentes criterios que se han tenido en cuenta para realizar el estudio comparativo. Los criterios se han separado en 2 grupos: a) los aspectos relacionados con el desarrollo de la RIA; y b) aquellas características centradas en el modelado de su IU. El primer grupo permite la clasificación de las propuestas desde dos puntos de vista: desde la naturaleza de su adaptación y desde el aspecto de la RIA en el que se centra. El segundo grupo se enfoca en analizar el modelado de la IU para una RIA. De esta manera, se puede conocer en qué aspectos coinciden y divergen las diferentes propuestas a la hora de modelar la IU.

#### 3.1 Criterios relacionados con los aspectos del desarrollo

Estos criterios analizan cómo las diferentes propuestas llevan a cabo el desarrollo de RIAs. A continuación, se presentarán los dos criterios utilizados y los valores que pueden tomar.

- Según el **enfoque de la propuesta**: las diferentes propuestas plantean enfoques diversos para la adaptación al desarrollo de RIAs. Según [23] se puede establecer una clasificación de estas propuestas teniendo en cuenta estos enfoques. A continuación se presentan los valores que este criterio puede tomar:
  - Extensión: algunas propuestas extienden un método ya existente para adaptarlo al desarrollo de RIAs.

- Combinación: cuando se propone combinar dos métodos diferentes para cubrir diferentes aspectos del desarrollo de RIAs.
- Nueva creación: son propuestas que se crean expresamente para el modelado e implementación de RIAs o de alguno de sus aspectos.
- Uso de patrones de comportamiento: Uso de patrones de comportamiento para definir el comportamiento de la IU.
- Definición de un metamodelo: Definición de un metamodelo que permite el modelado de IUs en RIA.
- Según el **aspecto central de la propuesta**: los 4 aspectos principales que caracterizan una RIA son: disponer de un cliente que pueda almacenar datos y ejecutar parte de la lógica de negocio, ofrecer una interfaz rica y permitir una comunicación asíncrona entre el cliente y el servidor [1]. Cada propuesta se centra en modelar uno o varios de estos aspectos, pudiendo establecer una nueva clasificación.
  - Distribución de los datos (Distr. Datos): la propuesta se centra en distribuir los datos entre el cliente y el servidor.
  - Distribución de la lógica de negocio (Distr. L.N): la propuesta dirige sus esfuerzos en modelar la distribución de la lógica de negocio entre el cliente y el servidor.
  - Modelado de la IU (Mod. IU): la propuesta se encauza en modelar las IU de la RIA.
  - Modelado de la comunicación (Mod. Com.): disponer de un cliente que pueda comunicarse de manera síncrona o asíncrona con el servidor.

### 3.2 Criterios para el modelado de la IU para RIAs

El aspecto del modelado de la IU es el aspecto de la RIA más abordado por las diferentes propuestas. Por esta razón, se ha considerado oportuno dedicar un apartado de criterios que ayuden a analizar cómo las propuestas llevan a cabo su modelado e implementación.

Dos conceptos importantes a tener en cuenta para el modelado de la IU son: a) el aspecto estático y b) el aspecto dinámico [10]. El aspecto estático define la estructura de la interfaz y los *widgets* que componen la interfaz. Por otro lado, el aspecto dinámico especifica el comportamiento de los elementos que la constituyen. Para nuestro estudio comparativo se proponen criterios para ambos aspectos y así poder analizarlos de forma separada.

Para el aspecto estático de la interfaz se establecen los siguientes criterios:

- Según los **Niveles de abstracción HCI**: en los últimos años la comunidad HCI (*Human Computer Interaction*) propone la utilización de niveles de abstracción para modelar el aspecto estático de la IU [10]. Sugiere utilizar un modelo independiente de plataforma (PIM, *Platform Independent Model*) para definir la estructura y los elementos de la interfaz, para luego extenderlo a una plataforma tecnológica en concreto (p. ej. Adobe Flex platform [24] u OpenLazslo [25]) y así obtener un modelo específico de plataforma (PSM, *Platform Specific Model*). Con este criterio se pretende analizar si la



propuesta tiene en cuenta la división en niveles que propone la comunidad HCI.

- Según el **modelo utilizado**: para la definición del aspecto estático cada propuesta presenta un modelo.
- Según el **nivel de abstracción del modelo utilizado**: Interesa saber qué nivel de abstracción tiene el modelo utilizado (PIM o PSM).
- **Definición del *look & feel***: si además de definir los elementos de la interfaz se especifican sus propiedades visuales (color, posición, dimensión, etc.)
- **Adaptación de la interfaz al dispositivo de salida**: si la propuesta tiene en cuenta las características del dispositivo de salida para estructurar la IU.

Referente al **aspecto dinámico** de la interfaz se pueden establecer los siguientes criterios:

- **Modelo utilizado**: el modelo utilizado para definir el aspecto dinámico.
- **Nivel de abstracción del modelo utilizado**: qué nivel de abstracción tiene el modelo que definirá el comportamiento dinámico de la IU (PIM o PSM).
- Según el **comportamiento modelado**: Se pueden distinguir 3 tipos de comportamiento que pueden ser desempeñados por un *widget*.
  - Interacción: un *widget* recibe un evento y ejecuta una acción.
  - Orquestación: un *widget* interactúa con otros *widgets* de la IU.
  - Cambios en la IU: el *widget* cambia su estado de visualización en la interfaz.
- **Representación gráfica del comportamiento**: la representación gráfica de los 3 tipos de comportamiento que se modelan se puede realizar mediante un único diagrama gráfico o utilizar uno diferente para cada tipo de comportamiento. Es decir, la representación puede ser conjunta o separada.

## 4 Estudio comparativo

Una vez se han presentado las diferentes propuestas y se han definido los criterios de evaluación, se hará el estudio comparativo. El estudio se presentará siguiendo los 2 grupos de criterios establecidos.

### 4.1 Estudio de las propuestas en función del aspecto de desarrollo

Siguiendo el criterio del **enfoque de la propuesta**, se comprueba que 3 de las propuestas estudiadas son extensiones de métodos ya existentes. Éstas son OOH4RIA, que extiende el método OOH [6], y las propuestas que plantean OOHDM [7] y WebML [4], ambas para cubrir diferentes aspectos de las RIAs.

Por otra parte, dos de las propuestas sugieren la combinación de 2 métodos diferentes. Una es la propuesta de WebML combinado con RUX Method y la otra la de UWE también combinada con RUX Method.

De las propuestas estudiadas en este trabajo, RUX Method [8] es la única que surge expresamente para el modelado de la RIAs.

Otra de las propuestas de UWE es la utilización de patrones de comportamiento para definir el aspecto dinámico de los *widgets* que componen la IU [20].

La última propuesta por clasificar es la de OOWS que se centra en crear un metamodelo que recoge tanto el aspecto estático como dinámico de la IU de las RIAs [10].

En cuanto al **aspecto central de la propuesta** se observa que la mayor parte de ellas centra sus esfuerzos en el modelado de la IU (OOH4RIA, OOHDM, RUX Method, UWE y OOWS). OOH4RIA lo hace mediante los modelos de presentación y orquestación [5], OOHDM utiliza el modelo de ADV y el modelo de ADV Chart [7], RUX Method propone la utilización del modelo de interfaz abstracta, concreta y final [8], UWE con la combinación con RUX Method [9] y la utilización de patrones para modelar el comportamiento de los *widgets* [20] y OOWS crea un metamodelo RIA [10].

Por otra parte, la propuesta de WebML se enfoca en los otros 3 aspectos principales de la RIA (el almacenamiento distribuido de los datos entre el cliente y el servidor, la distribución de la lógica de negocio y el modelado de la comunicación, síncrona y asíncrona, entre ambas partes) [4]. La distribución de los datos queda reflejada en el modelo conceptual de datos y la de la lógica de negocio en el modelo de navegación.

Por lo tanto, la propuesta que consigue abarcar los 4 aspectos de la RIA es la combinación de WebML con RUX Method [2].

A modo de resumen, la Tabla 1 recoge los resultados del análisis de las propuestas en relación a la estrategia de desarrollo escogida.

**Tabla 1.** Comparativa de propuestas en función de su estrategia de desarrollo.

Método	Propuesta	Enfoque de la propuesta	Aspecto Central
<b>OOH4RIA</b>	OO-H extendido	Extensión	Mod. IU
<b>OOHDM</b>	OOHDM extendido	Extensión	Mod. IU
<b>RUX</b>	RUX Method	Nueva creación	Mod. IU
<b>WebML</b>	WebML extendido	Extensión	Distr. Datos, Distr. L.N, Mod. Com
	WebML + RUX	Combinación	Distr. Datos, Distr. L.N, Mod. Com., Mod. IU
<b>UWE</b>	UWE + patrones	Uso de patrones	Mod. IU
	UWE + RUX	Combinación	Mod. IU
<b>OOWS</b>	OOWS + metamodelo	Definición metamodelo	Mod. IU

## 4.2 Estudio de las propuestas en función del modelado de la IU para RIAs

Respecto al **aspecto estático** de la IU se observa que la mayoría de propuestas definen la estructura de la interfaz mediante elementos contenedores que organizan y distribuyen el resto de elementos. El modelo de presentación de OOH4RIA utiliza los elementos *screenshot* para definir la estructura [5]. De igual forma, OOHDM lo hace mediante los elementos *ADV composite* en su modelo de ADVs [26]. RUX Method utiliza el elemento *view* para agrupar información en su modelo de interfaz abstracta [8]; UWE lo hace en su modelo de presentación mediante los

elementos *presentation class* [9], y el metamodelo RIA propuesto por OOWS recoge la entidad *WLayout* como *widget* que almacena a otros [10].

En cuanto al nivel de abstracción de los modelos, la mayoría establecen los dos niveles (PIM y PSM) que recomienda la comunidad HCI para el modelado de la IU [10]. Primero crean un modelo PIM que luego será extendido para una plataforma específica. A partir de este modelo específico se genera, normalmente, el código que implementa la interfaz. En el caso de RUX Method primero especifica los modelos de interfaz abstracta y concreta que son de nivel PIM para terminar con la definición del modelo de interfaz final que es PSM [8]. Por otra parte, el metamodelo RIA de OOWS se puede extender para crear un metamodelo específico para cada plataforma de implementación [10]. Sin embargo, las propuestas de OOH4RIA y OOHDM no hacen esta separación de niveles de abstracción. El modelo de presentación de OOH4RIA está definido directamente utilizando la plataforma GWT (*Google Web Toolkit*) [27] y el modelo de ADVs de OOHDM es un modelo PIM del cual se genera el código de forma directa.

La mayor parte de las propuestas especifica también el look & feel (dimensión, posición, estilo, etc.) de los *widgets*. Sin embargo, UWE no lo hace en su modelo de presentación.

En cuanto a la adaptación de la IU a las características del dispositivo de salida se ha comprobado que las dos únicas propuestas que lo tienen en cuenta son OOH4RIA con la introducción del modelo de usuario (*user model*) [12] y RUX en su modelo de interfaz concreta.

A modo de resumen, la Tabla 2 recoge los resultados del análisis del aspecto estático de la IU.

**Tabla 2.** Comparación con los criterios del aspecto estático de la IU para RIAs

Método	Propuesta	Niveles HCI	Modelo	Nivel abstracción modelo	Modelado del look & feel	Adaptación dispositivo salida
OOH4RIA	OO-H extendido	No	Presentación	PSM	Sí	Sí
OOHDM	OOHDM extendido	No	Modelo de ADVs	PIM	Sí	No
RUX	RUX Method	Sí	Interfaz abstracta	PIM	Sí	Sí
UWE	UWE + patrones	Sí	Presentación	PIM	No	No
OOWS	OOWS + metamodelo RIA	Sí	Metamodelo RIA estático	PIM	Sí	No

Para al **aspecto dinámico** de la IU OOH4RIA propone el modelo de orquestación de nivel PSM (plataforma GWT). Este modelo refleja los elementos *screenshot* y sólo aquellos *widgets* que ejecutan una operación después de recibir un evento; reciben el nombre de *orchestral widgets* [6]. Por otra parte, OOHDM propone el modelo ADV Chart de nivel PIM donde se representan los estados de visible o no visible de un ADV. Además, si se trata de un ADV contenedor (*ADV*

*composite*) se representan, dentro de su estado de visible, aquellos *subADV*s que también pasarán a ser visualizados [7]. RUX modela el aspecto dinámico de la IU con el modelo de interfaz concreta de nivel PIM donde separa dos conceptos importantes: la presentación temporal y la presentación de interacción de la IU [8]. UWE propone la utilización de patrones de nivel PIM para definir el comportamiento de los *widgets* que componen la interfaz [20]. OOWS también recoge el aspecto dinámico de la IU en su metamodelo RIA. Para ello, incluye las entidades *Event*, *EventRule* y *Reaction* [10].

En cuanto a los tipos de comportamiento se observa que todas las propuestas definen el comportamiento de interacción de los *widgets*. Es decir, todas modelan la reacción de un *widget* al recibir un evento lanzado, normalmente, por el usuario. Las interacciones entre *widgets* (el comportamiento de orquestación) son modeladas por todas las propuestas exceptuando UWE. Por otra parte, los cambios en la interfaz sólo son modelados por la propuesta de OOHDM.

Los diagramas de estados son la técnica más utilizada para la representación gráfica de los modelos. La mayoría de las propuestas utilizan un solo diagrama para representar los diferentes comportamientos de los *widgets*. Sólo RUX utiliza dos diagramas distintos; un diagrama de estados para la presentación de interacción (comportamiento de interacción) y un diagrama de secuencia para la presentación temporal (comportamiento de orquestación).

A modo de resumen, la Tabla 3 recoge los resultados del análisis del aspecto dinámico de la IU.

**Tabla 3.** Comparación con los criterios del aspecto dinámico de la IU para RIAs

Método	Propuesta	Modelo	Nivel abstracción modelo	Comportamiento modelo	Representación gráfica
OOH4RIA	OO-H extendido	Orquestación	PSM	Interacción y orquestación	Conjunta
OOHDM	OOHDM extendido	Modelo de ADVs Chart	PIM	Interacción y orquestación	Conjunta
RUX	RUX Method	Interfaz concreta	PIM	Interacción, orquestación, cambios en la IU	Separada
UWE	UWE + patrones	Patrones	PIM	Interacción	Conjunta
OOWS	OOWS + metamodelo RIA	Metamodelo RIA dinámico	PIM	Interacción y orquestación	Conjunta

## 4 Conclusiones

La aparición de las RIAs ha tenido una importante incidencia sobre el proceso de desarrollo de las aplicaciones Web. Los métodos dirigidos por modelos que hasta ahora modelaban e implementaban aplicaciones Web tradicionales, han visto la necesidad de adaptarse al nuevo de desarrollo de RIAs.

Las características más importantes de estas nuevas aplicaciones Web son: la de dotar a la parte del cliente de la capacidad de almacenar datos y de ejecutar parte de la lógica de negocio, la de presentar interfaces de usuario con un alto grado de interactividad y usabilidad y poder comunicarse con el servidor de manera síncrona o asíncrona.

Con este estudio comparativo se comprueba que la mayoría de las propuestas presentadas en este trabajo han enfocado sus esfuerzos en adaptar el modelado de la IU al nuevo paradigma RIA.

De esta manera, se concluye que para el modelado de estas interfaces es necesario tener en cuenta dos aspectos: el aspecto estático y dinámico. El aspecto estático es heredado del aspecto de la presentación que definía la estructura de las IUs de las Webs tradicionales. En cambio, el aspecto dinámico, que permite modelar el comportamiento de los elementos de la interfaz, ha sido introducido para el desarrollo de IUs para RIAs. Para terminar, es interesante comentar que es en el modelado de este aspecto donde más difieren las propuestas analizadas.

## 5 Agradecimientos

Este trabajo se ha llevado a cabo en el marco de los proyectos MODEL-CAOS (Ref. TIN2008-03582) y Agreement Technologies (CONSOLIDER CSD2007-0022) financiado por el Ministerio de Ciencia y Tecnología de España.

## Referencias

- [1] Toffetti, G., Comai, S., Preciado, J. C. and Linaje, M. State-of-the-art and trends in the Systematic Development of Rich Internet Applications. *Journal of Web Engineering*, 10, 1 (2011), 70-86.
- [2] Brambilla, M., Preciado, J. C., Linaje, M. and Sanchez-Figueroa, F. Business Process - based Conceptual Design of Rich Internet Applications. In *Proceedings of the International Conference on Web Engineering* (Yorktown Heights, NJ 2008).
- [3] Fraternali, P., Comai, S., Bozzon, A. and G., T. Engineering Rich Internet Applications with a Model-Driven Approach. *ACM Transactions on the web*, 4, (2010), 47.
- [4] Bozzon, A., Comai, S., Fraternali, P. and Toffetti, G. Conceptual Modeling and Code Generation for Rich Internet Applications. In *Proceedings of the International Conference on Web Engineering* (2006). ACM New York.
- [5] Meliá, S., Gómez, J., Pérez, S. and Díaz, O. A model-driven development for GWT-based Rich Internet applications with OOH4RIA. In *Proceedings of the International Conference on Web Engineering* (Yorktown Heights, 14-18 July 2008).
- [6] Pérez, S., Díaz, O., Meliá, S. and Gómez, J. Facing Interaction-Rich RIAs: the Orchestration Model. In *Proceedings of the International Conference on Web Engineering* (14-18 July 2008).
- [7] Urbieto, M., Rossi, G., Ginzburg, J. and Schwabe, D. Designing the Interface of Rich Internet Applications. In *Proceedings of the Latin American Web Conference* (Santiago 2007). IEEE Computer Society Washington.

- [8] Linaje, M., Preciado, J. C. and Sanchez-Figueroa, F. A method for Model Based Design of Rich Internet Application Interactive User Interfaces. In *Proceedings of the International Conference Web Engineering* (2007).
- [9] Preciado, J. C., Linaje, M., Sánchez-Figueroa, F., Morales-Chaparro, R., Zhang, G., Kroiss, C. and Koch, N. Designing Rich Internet Applications combining UWE and RUX-Method. In *Proceedings of the International Conference on Web Engineering* (2008). IEEE Computer Society Washington.
- [10] Valverde, F. and Pastor, O. Facing the Technological Challenges of Web 2.0: a RIA Model-driven Engineering Approach. In *Proceedings of the International Conference on Web Engineering* (2009).
- [11] Gómez, J. and Cachero, C. OO-H Method: Extending UML to model web interfaces. *Information Modelling for Internet Applications*, Idea Group Publishing, 2003).
- [12] Garrigós, I., Meliá, S. and Casteleyn, S. Adapting the presentation layer in Rich Internet Applications. In *Proceedings of the International Conference on Web Engineering* (2009). Springer-Verlag, Berlin.
- [13] Schwabe, D. and Rossi, G. An Object Oriented Approach to Web-Based Application Design. *Theory and Practice of Object Systems*(October 1998 1998), 207-225.
- [14] Cowan, D. and Pereira de Lucena, C. Abstract Data Views: An Interface Specification Concept to Enhance Design for Reuse. In *Proceedings of the IEEE Trans. Software Engineering* (1995).
- [15] Wohed, P., van der Aalst, W. M. P., Dumas, M., ter Hofstede, A. H. M. and Russell, N. On the Suitability of BPMN for Business Process Modelling. *Lecture Notes in Computer Science*, 41022006), 161-176.
- [16] Preciado, J. C., Linaje, M., Comai, S. and Sánchez-Figueroa, F. Designing Rich Internet Applications with Web Engineering Methodologies. In *Proceedings of the International Symposium on Web Site Evolution* (5-6 Oct. 2007).
- [17] Pemberton, S., . and et al. *XHTML; 1.0 The Extensible HyperText Markup Language*. W3C, 2000.
- [18] Koch, N., Knapp, A., Zhang, G. and Baumeister, H. UML-Based Web Engineering: An Approach Based on Standards. *Web Engineering: Modelling and Implementing Web Applications*, 12, 2007), 157-191.
- [19] Scott, B. RIA Patterns. Best Practices for Common Patterns of Rich Interaction ([www.uxmatters.com/mt/archives/2007/03/](http://www.uxmatters.com/mt/archives/2007/03/)).
- [20] Koch, N., Pigerl, M., Zhang, G. and Morozova, T. Patterns for the Model-Based Development of RIAs. In *Proceedings of the International Conference on Web Engineering* (2009). Springer-Verlag Berlin.
- [21] Fons, J., Pelechano, V., Albert, M. and Pastor, O. Development of Web Applications from Web Enhanced Conceptual Schemas. *Lecture Notes in Computer Science*, 2813, 2003), 232-245.
- [22] Cicchetti, A., Di Ruscio, D. and Pierantonio, A. Weaving Concerns in Model Based Development of Data-Intensive Web Applications. In *Proceedings of the Symposium on Applied Computing* (2006).
- [23] Busch, M. and Koch, N. *Rich Internet Applications. State of the art*. Technical report, Ludwig-Maximilians-Universität München, , München, Germany, 2009.
- [24] *Adobe Flex Developer Center* (<http://flex.org/>)
- [25] *Openlaszlo* ([www.openlaszlo.org](http://www.openlaszlo.org)).
- [26] Rossi, G., Urbietta, M., Ginzburg, J., Distanto, D. and Garrido, A. Refactoring to Rich Internet Applications. A Model-Driven Approach. In *Proceedings of the International Conference on Web Engineering* (Yorktown Heights, NJ 14-18 July 2008).
- [27] *Google Web Toolkit* (<http://code.google.com/intl/ca/webtoolkit/>).