

# Mejora de la interoperabilidad de las RIAs desde una perspectiva semántica: Un caso de estudio en la Web Social

Jesús M. Hermida, Santiago Meliá, Andrés Montoyo, Jaime Gómez

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante  
Apto. de correos 99, E-03080, Alicante, Spain  
{jesusmhc, santi, montoyo, jgomez}@dlsi.ua.es

**Resumen.** En la Web 2.0, actualmente existe un problema de interoperabilidad cuando se intenta conectar y transmitir datos entre diferentes aplicaciones enriquecidas de internet (RIA), o entre RIAs y fuentes de datos. Este hecho dificulta la reutilización de información entre diferentes fuentes de datos y en algunos casos hace que esta información quede “inmovilizada” o no reusable para otros sistemas. Cuando el valor de negocio recae en los datos y no en la aplicación, como ocurre en el entorno actual, es importante que la información fluya sin impedimentos y para ello las tecnologías de la Web semántica pueden ser clave en la solución. En concreto, este artículo presenta una solución desde la perspectiva de la metodología de diseño dirigida por modelos S<sup>m</sup>4RIA, empleada en el desarrollo de RIAs *semánticas*. Como caso de estudio, se aborda el modelado de una red social en el dominio de la música. Esta aproximación combina las principales ventajas presentes en las tecnologías RIA actuales y acelera el desarrollo de este tipo de aplicaciones simplificando la reutilización de fuentes externas de conocimiento.

*Palabras clave:* S<sup>m</sup>4RIA, RIA semántica, diseño basado en modelos

## 1 Introducción

La Web se encuentra en un proceso de cambio continuo hacia la llamada Web 3.0, o “Web Semántica Social” [1]. Durante la última década, las aplicaciones de la Web 2.0 [2] han promocionado el uso de la Web entre una gran variedad de usuarios. Dentro del amplio concepto “Web 2.0”, pueden ser identificadas al menos dos ramas principales, una asociada a los cambios en el comportamiento de los usuarios y otra referente a las aplicaciones ricas de Internet. Al mismo tiempo, las aplicaciones enriquecidas de Internet (RIAs) han producido un cambio en la forma de interacción y comunicación entre usuarios y aplicaciones, y a su vez entre las propias aplicaciones. Las RIAs ofrecen interfaces de usuario con un alto grado de interactividad, similar a las aplicaciones de escritorio, incluyen contenido multimedia y minimizan la comunicación entre los componentes cliente y servidor.

Ambas ramas de la Web 2.0 comparten algunos inconvenientes que limitan principalmente la portabilidad de los datos entre aplicaciones y, por tanto, la interoperabilidad entre ellas. Por una parte, los sitios sociales son creados como sitios propietarios que ofrecen un acceso a sus datos limitado por APIs cuya semántica puede variar dependiendo del sitio Web. Por ello, en la actualidad, es corriente que los usuarios tengan diferentes perfiles en diferentes redes ya que no pueden compartir los datos personales de una a otra (según Breslin et al. en [3] y [4]). Por otra parte, debido a cuestiones tecnológicas, las RIAs actúan como cajas negras que muestran el contenido de una forma amigable pero limitan el acceso a los datos sólo a los usuarios humanos, ya sean RIAs orientadas al navegador (cuyos datos se muestran en función de los eventos que los usuarios lanzan en la interfaz) o a los plugins (que son objetos binarios cuya información es únicamente visualizable mediante una extensión del navegador específica para cada tecnología de implementación).

En el contexto actual de Internet, donde el valor se ha desplazado de las aplicaciones Web a los propios datos que manejan, el uso de soluciones tecnológicas abiertas es una necesidad. El objetivo de la Web Semántica [5] es resolver los problemas de incompatibilidad semántica de sistemas por medio de técnicas y tecnologías estandarizadas. Éstas son la clave para solventar las cuestiones mencionadas y desarrollar la Web a un nuevo estado, el de los sistemas que contengan nuestra inteligencia colectiva [6]. La Web Semántica Social, como combinación de la Web Semántica y la Web Social está en proceso de ser una realidad con el desarrollo de nuevos tipos de aplicaciones como los blogs y los wikis semánticos (descritos por Kinsella et al. [7]), y diferentes aproximaciones con el objetivo de reutilizar el conocimiento contenido en las diferentes plataformas sociales, tales como FOAF (en inglés "*Friend of a Friend*", "Amigo de un Amigo").

Tras años de desarrollo estas tecnologías, técnicas y aplicaciones son lo suficientemente maduras como para ser tratadas desde la perspectiva de la ingeniería Web. Dentro el contexto descrito, este artículo presenta una solución basada en S<sup>m</sup>4RIA (en inglés "*Semantic Models for RIA*", Modelos semánticos para RIA), una metodología de diseño basado en modelos que permite la creación de RIAs semánticas, que facilitan la interconexión entre sistemas RIA y entre RIAs y otras fuentes de datos por medio de tecnologías de la Web semántica. En este caso, el artículo desarrolla como caso de estudio el diseño de un sitio Web social usando una RIA semántica como plataforma con la finalidad de explicar las diferentes fases de la metodología y los beneficios del uso de las tecnologías de la Web Semántica centrándose en las actividades que modelan la interoperabilidad entre aplicaciones.

Este artículo se encuentra dividido en los siguientes apartados: Los apartados 2 y 3 describen las principales características de las RIAs semánticas y presentan el caso de estudio utilizado en el artículo, respectivamente. El apartado 4 presenta el método S<sup>m</sup>4RIA y el apartado 5 describe cómo modelar la parte servidora de una SRIA haciendo énfasis en las tareas creadas para mejorar la interoperabilidad de las SRIA respecto de la RIA tradicional. Por último, los apartados 6 y 7 analizan de los trabajos previos y presentan las conclusiones del trabajo así como las líneas de trabajo futuro.

## 2 RIAs semánticas como plataforma: Estructura y funcionalidad

A pesar de los beneficios de usar RIAs, estas aplicaciones comparten un inconveniente con los sitios Web sociales: la exportabilidad y reusabilidad de los datos que contienen. Actualmente existen algunas soluciones disponibles que explotan el contenido del código HTML donde se encuentran incrustadas las RIAs orientadas al navegador, como se viene haciendo en la Web tradicional. Sin embargo, todas estas aproximaciones son dependientes de la tecnología de implementación de la RIA.

La aproximación que se utiliza en este artículo está basada en el concepto de RIA semántica (SRIA), definida como una RIA que hace uso intensivo de las tecnologías de la Web Semántica para ofrecer una representación de sus contenidos y reutilizar las fuentes de conocimiento ya existentes en la Web (ver [8] para una descripción más detallada de sus requisitos).

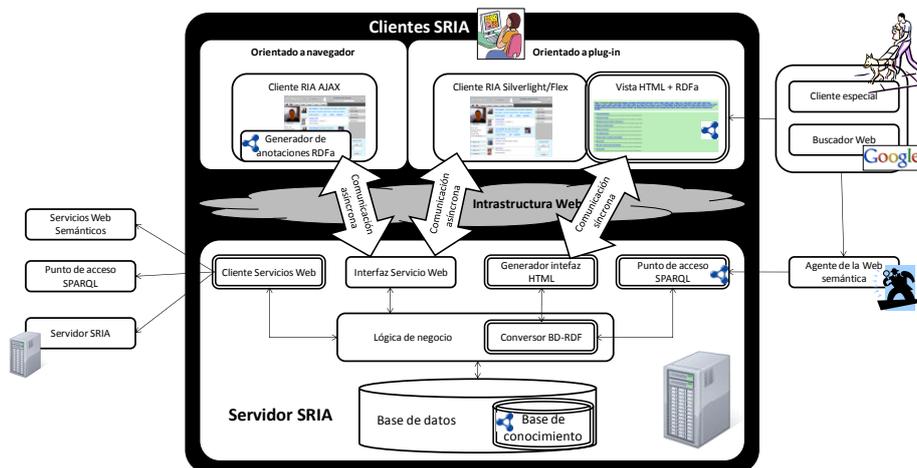


Fig. 1. Esquema básico de una RIA semántica.

La figura 1 presenta un esquema conceptual de una SRIA con sus relaciones con los usuarios y otros componentes de la Web. Se ilustra la SRIA como una caja negra, como sería una RIA también, con una arquitectura cliente-servidor, cuyos bloques comparten datos mediante procesos de comunicación asíncronos dentro de la infraestructura de la Web. No obstante, las SRIA contienen cuatro módulos software (con doble borde en la figura 1) que no están tradicionalmente presentes en las RIAs:

- **Base de conocimiento (KB)**, *servidor*. Este módulo gestiona la base de conocimiento de la aplicación, que incluye las ontologías que representan el contenido de la aplicación y las reglas de mapeo entre los elementos de la BD y las instancias de esas ontologías. Se acompaña de un módulo (convertor BD-RDF) que convierte las instancias de datos a RDF acorde a las reglas de mapeo.
- **Punto de acceso SPARQL**, *servidor*. Este módulo ofrece un servicio para acceder a parte del conocimiento almacenado en la KB de la RIA. En este caso

la aproximación está alineada con los principios de Linked Data (definidos por Bizer et al. [9]) debido a la simplicidad de los procesos que define y el protocolo SPARQL (<http://www.w3.org/TR/rdf-sparql-protocol/>). Sin embargo, esta interfaz podría ser reemplazada dependiendo de los requerimientos de la aplicación.

- **Ciente de los servicios Web (semánticos), servidor.** Este cliente es una combinación de diferentes clientes que pueden acceder a diversos tipos de contenidos y servicios en la Web. El módulo facilita el acceso de la aplicación a las ontologías y KB disponibles según requiera.
- **Vista HTML, cliente.** Para poder acceder al contenido de aquellas RIAs que requieren de extensiones en el navegador Web (plugins), las SRIA incorporan una vista HTML de sus datos, anotado todo con RDFa. El modelo de anotación está basado en dos tipos de anotaciones semánticas: de contenido y de contexto. Las primeras están relacionadas directamente con el dominio (y la finalidad) de la aplicación, mientras que las segundas ofrecen conocimiento sobre el contexto de navegación y los elementos de visualización, que puede ayudar a interpretar el contenido presentado en la aplicación. Los clientes para usuarios con algún tipo de impedimento físico, tales como usuarios con deficiencias visuales, serían el objetivo ideal este tipo de anotaciones. Además de utilizar la vista HTML, los usuarios podrán acceder al mismo conocimiento utilizando la interfaz SPARQL.
- **Generador de anotaciones (semánticas), cliente.** El desarrollo de este módulo depende de la tecnología de implementación de la RIA. En aquellos casos en que la SRIA se implementa usando tecnologías basadas en Javascript, es posible incluir un componente software que incluya dentro del código HTML anotaciones basadas en RDFa, las cuales apuntarán a instancias de ontologías (o a las propias ontologías) almacenadas en la KB.

Las RIAs semánticas son una plataforma genérica para el desarrollo de diferentes tipos de aplicaciones que ofrecen una solución independiente de la tecnología de implementación utilizada. Los principales beneficios de utilizar las SRIAs como una plataforma son las siguientes:

1. *Mejora en la interoperabilidad entre aplicaciones.* El uso de ontologías permite a los diseñadores describir todos los datos de una forma estándar y facilita el proceso de compartir datos entre diferentes sitios Web. Por ejemplo, la ontología FOAF es ampliamente utilizada en la Web para describir personas, documentos y sus relaciones.
2. *Enriquecimiento de los contenidos.* SRIAs pueden reutilizar el contenido de otros sitios Web o fuentes de conocimiento existentes para enriquecer el contenido presentado a los usuarios. Por ejemplo, una SRIA podría automáticamente acceder al contenido de DBpedia (<http://www.dbpedia.org>) para recuperar más información sobre un determinado tema. Además, las SRIAs pueden contener anotaciones semánticas en el contenido HTML visualizado por los usuarios, las cuales podrán ser utilizadas por diferentes tipos de usuarios, como buscadores Web o clientes para usuarios con necesidades especiales o con algún tipo de discapacidad.

### 3 Caso de estudio: Red social en una RIA semántica

En este artículo, el caso de estudio que se abordará es el desarrollo de una red social utilizando como plataforma una SRIA (ver Fig. 2). La red social que se creará no será general sino que estará enfocada en el dominio musical. Además incluirá un reproductor musical para que los usuarios puedan escuchar sus canciones favoritas, siendo así una potencial plataforma de venta de música.

Además de incluir las principales funcionalidades de las redes sociales actuales (gestión del perfil de usuario, conexiones con otros usuarios y un *muro* donde expresan pensamientos, impresiones y comentarios con sus amigos), los usuarios podrán compartir sus preferencias musicales (grupos y canciones), seguir el muro de sus artistas favoritos y leer y publicar noticias de todos estos temas. La aplicación final no incluye nuevas características en cuanto al aspecto social, que ya han sido explotadas por algunas redes sociales como Ping (<http://www.apple.com/itunes/ping/>) o lastFM (<http://www.last.fm/>). Este caso de estudio está centrado en dos cuestiones: la interconexión entre redes sociales utilizando técnicas y recursos de la Web Semántica ya estudiados en otros trabajos [7], y el análisis del proceso de desarrollo descrito en el próximo apartado. Las cuestiones referentes a la seguridad de la aplicación, privacidad de los usuarios y confiabilidad de las fuentes quedan fuera del ámbito del artículo.

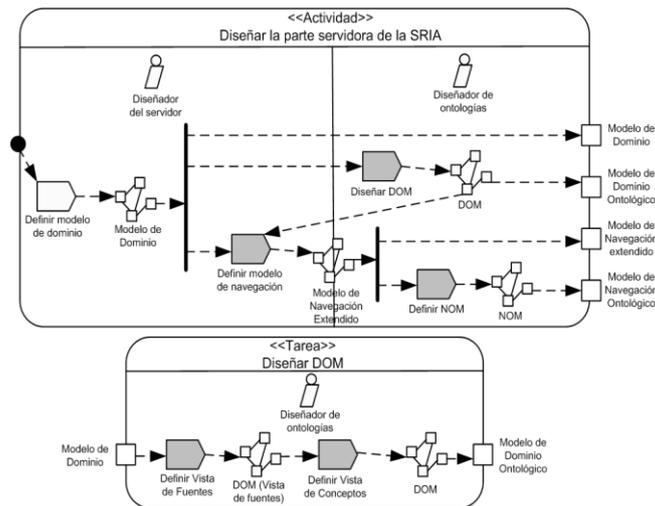


Fig. 2. Captura de pantalla de la red social propuesta.

### 4 Introducción a la metodología S<sup>m</sup>4RIA

Este apartado presenta la metodología S<sup>m</sup>4RIA, que extiende el proceso original OOH4RIA [10] modificando las tareas existentes e introduciendo una colección de nuevas tareas. El proceso de desarrollo puede ser dividido en tres actividades, que

agrupan procesos con la misma finalidad: diseñar los componentes de la parte servidora de la SRIA, diseñar los componentes de la parte cliente de la SRIA y, por último, generar la SRIA por medio de un conjunto de procesos de transformación modelo a texto.



**Fig. 3.** Diagrama SPEM de la actividad “*Diseñar la parte servidora de la SRIA*”.

La primera actividad está representada en la figura 3 mediante un diagrama SPEM (<http://www.omg.org/spec/SPEM/2.0/>), en el cual las actividades de color gris son aquellas definidas o modificadas por S<sup>m</sup>4RIA respecto de OOH4RIA. Esta comienza cuando el diseñador del servidor define el modelo de Dominio, que incluye la información necesaria para especificar el diseño de la parte servidora de una RIA con las operaciones CRUD (crear, leer, actualizar y eliminar, en inglés, *Create, Read, Update, Delete*). A partir de este modelo, el diseñador de ontologías define las diferentes relaciones semánticas con las fuentes de ontologías externas de tal forma que los conceptos incluidos en la SRIA pueden ser alineados con los conceptos de otras fuentes o aplicaciones. Como resultado de esta tarea, se obtiene el modelo ontológico de dominio (en inglés, *Domain Ontology model, DOM*), que es a su vez un prerequisite para la siguiente tarea: “Definir modelo de Navegación”. En esta tarea, el diseñador define cómo el usuario será capaz de navegar por los datos de la SRIA por medio de un conjunto de clases navegacionales, que referencian los conceptos definidos en los modelos de dominio. Por medio del modelo de Navegación extendido los diseñadores pueden definir el acceso a bases de conocimiento, utilizando consultas OCL, y el uso de información de las mismas dentro de la SRIA. El último modelo definido en esta actividad es el modelo ontológico de navegación (en inglés, *Navigation Ontology model, NOM*), que captura cómo las SRIAs publican su información y conectan su información con la de otras fuentes de la Web.

La segunda actividad continua con la transformación del modelo de navegación extendido en el modelo de Presentación y en el modelo de Orquestación en dos pasos

consecutivos por medio de dos transformaciones llamadas *Nav2Pres* y *NavPres2Orq* (ver [10] para más detalle). El modelo de Presentación representa la estructura de los elementos de la interfaz de la SRIA que es completada por el modelo de Orquestación, que representa la interacción entre los elementos definidos en el modelo de Presentación y los elementos del modelo de Navegación Extendido.

Finalmente, la última actividad del método tiene como meta la generación de la SRIA a partir del conjunto de modelos creados en las dos primeras actividades por medio de un conjunto de procesos de transformación modelo a texto, que obtienen el código de cada uno de los componentes software descritos en el apartado 2.

## 5 Diseño de los componentes de servidor de la red social

El presente artículo se centrará exclusivamente en la descripción de la primera actividad del método, en la cual los diseñadores de la aplicación definen la interconexión entre aplicaciones, la reutilización de fuentes de datos externas, y establece qué información será compartida por la SRIA por medio de sus ontologías.

### 5.1 Modelado conceptual del dominio

El modelado conceptual de los datos de la aplicación se realiza mediante una aproximación híbrida que combina dos modelos complementarios: el modelo de dominio y el modelo ontológico de dominio. Ambos modelos permiten representar las estructuras de datos con las que trabajará la SRIA. El modelo de dominio permite definir las estructuras de datos de la aplicación así como su comportamiento en ejecución, establece las principales acciones sobre los datos (crear, modificar, leer y eliminar) y define cómo manejar los datos para su almacenamiento persistente. El modelo ontológico de dominio permite definir estas estructuras como una ontología indicando que conceptos serán públicos (es decir, se generarán dentro de la ontología de dominio) y permitiendo la alineación de los conceptos del dominio con ontologías ya existentes, y así reutilizar conocimiento ya definido en otras fuentes.

La primera actividad del proceso S<sup>m</sup>4RIA comienza cuando el diseñador del servidor crea el modelo de dominio de la aplicación, representado en la Fig. 4, que en este caso representa los conceptos gestionados internamente por la red social. El esquema que se presenta está dividido en dos partes: una social, que sería similar en cualquier red social y que contiene las clases necesarias para la interacción social de una comunidad de usuarios y otra dependiente del dominio de la aplicación, que en este caso está asociado con el dominio musical.

En esta red social existen dos tipos de usuarios: miembros de la comunidad (clase *User*, usuario) y miembros corporativos (clase *Artist*, artista). Los primeros son aquellos que interactúan, producen y consumen los principales contenidos de la red. Los artistas tienen también la posibilidad de gestionar un perfil oficial para interactuar con los seguidores y fans. Todos los miembros de la red poseen un “muro” donde poder publicar historias (clase *Story*, historia) y comentar (clase *Comment*, comentario) sus propias historias o las de sus amigos. Asimismo, los

usuarios pueden enviar mensajes privados entre ellos (clase *Message*, mensaje). Como la red esta centrada en el dominio musical, cada usuario puede crear sus propios álbumes (clase *UserAlbum*, álbum de usuario) a partir de una lista de canciones (clase *Song*, canción) propias, que pueden ser etiquetadas usando etiquetas propias (clase *Tag*, etiqueta) o de otros usuarios.

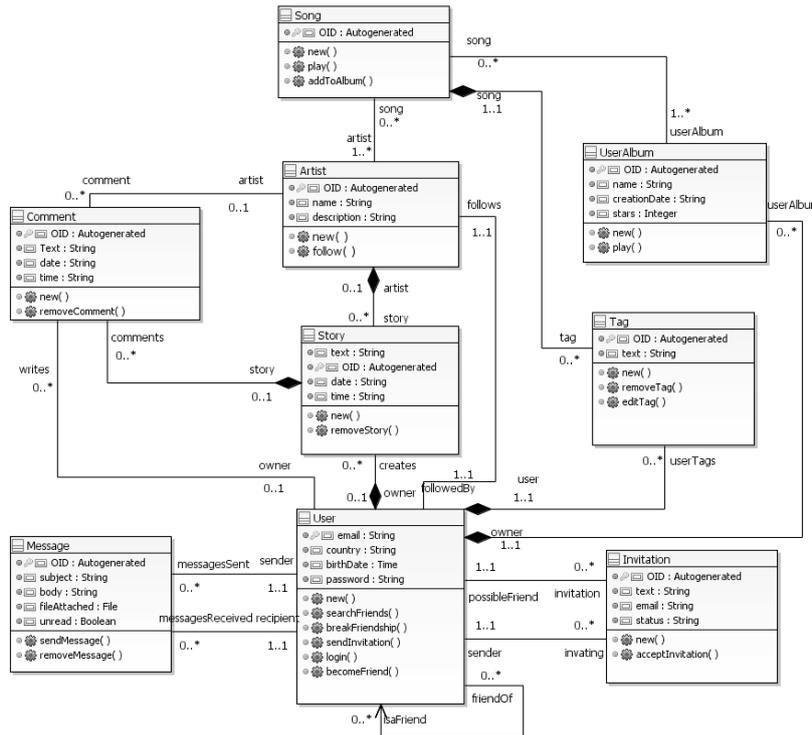


Fig. 4. Modelo de dominio para el caso de estudio.

Una vez completado el modelo de Dominio, el diseñador de la ontología construye el modelo ontológico de dominio, es decir, la ontología de dominio de la aplicación, que puede enlazar los conceptos de la aplicación con otras ontologías y establece qué fuentes de conocimiento (ontologías y sus instancias) serán utilizadas. Este modelo está basado en el metamodelo OntoDom, que extiende el metamodelo ODM (en inglés, *Ontology Definition Metamodel* [11], metamodelo de definición de ontologías). Este metamodelo simplifica la definición de elementos de la ontología e incluye un nuevo elemento, la fuente de instancias de ontología, como elemento *DOSource* (*Domain Ontology Source*), concepto que representa fuentes de conocimiento con diferente acceso: servicios Web semánticos, puntos de acceso SPARQL, etc. En un primer paso el diseñador define la vista de fuentes (Fig. 5) indicando qué ontologías van a ser importadas (como elementos *DODefinition*) y dónde se almacenan sus instancias (representado por elementos *DOSource*). En este

ejemplo el DOM importa dos ontologías, los elementos MusicOntology y FOAF, cuyas instancias están almacenadas en los elementos MusicBrainz y SNS\_FaceRIA (otra aplicación con un punto de acceso SPARQL), de tipo *DOSource*.

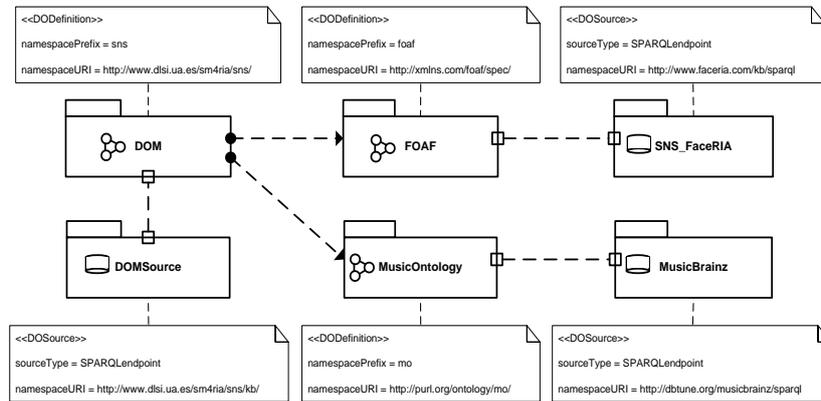


Fig. 5. Vista de fuentes del modelo ontológico de dominio para el caso de estudio.

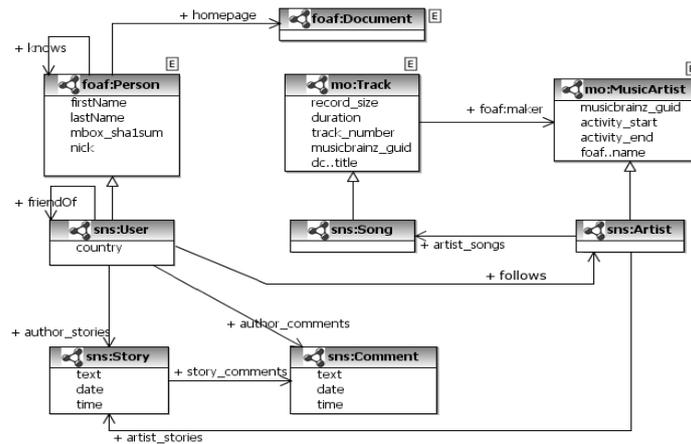


Fig. 6. Vista de conceptos del modelo ontológico de dominio para el caso de estudio.

Tras completar la vista, el diseñador de ontologías construye la vista de concepto del DOM, representado en la Fig. 6, en la que realmente construye la ontología de dominio de la aplicación (que está almacenada en la definición dentro del elemento DOM, de tipo *DODefinición*) y enlaza los conceptos que contiene con conceptos de las ontologías importadas. Para cada una de las clases del modelo de dominio el diseñador puede crear una única clase de ontología (*DOClass*), indicando sus propiedades y relaciones con otras clases de la ontología (también importados del modelo de dominio).

Siguiendo con el ejemplo, en este caso el diseñador ha extendido los conceptos *User* y *Song* con atributos de la clase *Person* de la ontología FOAF y de la clase

Track de la ontología MusicOntology respectivamente (los espacios de nombres se definen en la vista de fuentes, ver Fig. 5). Esto permitirá la generación de la información de la red social como instancias de las ontologías FOAF y MusicOntology facilitando así el intercambio de contenidos entre redes sociales. En el diagrama de la Fig. 6, aquellas clases de otras ontologías (*DODefinition* en la Vista de Fuentes) son destacadas como externas por medio del indicador “E” y los identificadores son agrupados en diferentes espacios de nombres.

## 5.2 Diseño de la navegación por los datos.

La siguiente tarea comprende el diseño el modelo de Navegación extendido, que extiende el modelo de navegación de OOH4RIA [10] y, representa cómo los usuarios navegarán por los datos contenidos en la aplicación. En la Fig. 7 se representan dos tipos de clases navegacionales en función de la clase del dominio de la que fueron creadas: locales, coloreadas en gris y creadas desde una clase del modelo de dominio; y externas, coloreadas en blanco y asociadas con una clase del DOM.

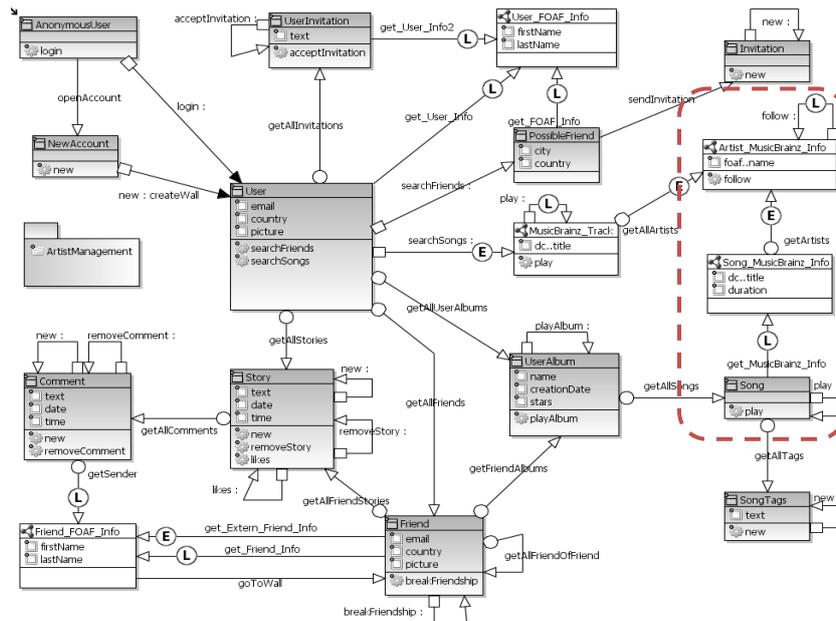


Fig. 7. Modelo de navegación extendido del caso de estudio.

Para acceder al muro, los usuarios tienen que registrarse primero y abrir un perfil, tras lo cual, la aplicación dará acceso a diferentes áreas. La primera de estas áreas incluye el muro del perfil, que contiene las últimas historias y comentarios (clases navegacionales *Story*, *Comment*, *Friend\_FOAF\_Info* – Historia, Comentario y Amigo\_FOAF\_Info) y una lista de sus amigos (clase navegacional *Friend* – Amigo), cada uno con un enlace a su propio muro. Un usuario puede buscar e invitar otros

usuarios para que formen parte de su núcleo de amigos (clases navegacionales *PossibleFriend*, *Invitation* – PosibleAmigo, Invitación) o aceptar las invitaciones que ha recibido (clase navegacional *UserInvitation* – InvitaciónRecibida). En cuanto a las funcionalidades dependientes del dominio musical, un usuario puede gestionar sus propios álbumes personales (y reproducirlos, clase *UserAlbum* – AlbumDeUsuario) y etiquetar las canciones que contienen (clases *Song*, *Canción\_MusicBrainz\_Info*, *Tag* – Canción, Canción\_MusicBrainz\_Info, Etiqueta). Igualmente es posible buscar y reproducir las canciones de MusicBrainz (clases *MusicBrainz\_Track*, *Artist\_MusicBrainz\_Info*). Las clases para gestionar los perfiles corporativos, en este caso los artistas, están incluidos en el paquete *ArtistManagement* – GestiónArtistas.

Los enlaces trasversales con destino una clase navegacional externa están obligatoriamente asociados a una única fuente de conocimiento. Para acceder a una clase navegacional externa, el diseñador escogerá la fuente de conocimiento asociada, que tiene que haber sido previamente definida en el DOM (mediante elementos *DOSource*), desde la cual se extraerán las instancias de la clase ontológica asociada a la clase navegacional. De esta forma establecerá el tipo de enlace: local (destino fuente de conocimiento local) o externo – enlaces “L” o “E”). Por ejemplo (ver cuadro con líneas discontinuas), el enlace *get\_MusicBrainz\_Info* está asociado a la fuente *DOSource* (base de conocimiento local a la SRIA) mientras que el enlace *get\_Artists* esta enlazado con la base de conocimiento de MusicBrainz.

Una vez completado, el diseñador de ontologías crea el modelo ontológico de navegación (NOM), que define una ontología con conceptos de navegación que 1) representará un mapa de navegación por la aplicación para los usuarios finales, 2) dónde se almacenan las instancias, es decir, en qué elemento *DOSource* local, y 3) restringe las instancias de la ontología de dominio que serán generadas por cada una de las clases navegacionales por medio de condiciones OCL. Para todo ello, utiliza un metamodelo propio, *OntoNavi*, basado en ODM, que contiene cuatro elementos principales: clases, enlaces, atributos y condiciones.

## 6 Trabajos relacionados

El uso de tecnologías de la Web semántica en las RIA de forma general es un campo poco explorado en la actualidad. Hasta donde llega el análisis realizado, no se han encontrado propuestas similares en la bibliografía que utilicen estas tecnologías en la extensión de las RIAs actuales. Por otra parte y como es de suponer, la extensión de los métodos de diseño dirigidos ya existentes para desarrollar RIA tampoco ha sido tratada en profundidad, aunque algunos pasos se han llevado a cabo desde el campo de la ingeniería Web. El propósito de este apartado es analizar en un primer paso las diferentes soluciones adoptadas por las metodologías más conocidas para el desarrollo de aplicaciones RIA y aplicaciones Web semánticas y, en un segundo paso, compararlas con la aproximación seguida en S<sup>m</sup>4RIA. Estas metodologías combinan mecanismos para modelar parte de los requerimientos de las aplicaciones SRIA. El análisis comenzará por aquellos métodos que mezclan primitivas para diseñar RIAs y el uso de tecnologías de la Web semántica.

El primero método a analizar es WebML [12], que fue el primero en incluir primitivas para modelar tanto interfaces de usuario enriquecidas y aplicaciones Web semánticas (entre otras aplicaciones, ver Ceri et al. [13]). Brambilla y Facca [14] definen las características y los requisitos de un portal Web semántico y proponen una extensión de WebML para desarrollar este tipo de aplicaciones. Al mismo tiempo, Brambilla et al. [15] establecieron un conjunto de requerimientos para el acceso a servicios Web semánticos desde aplicaciones Web. Sin embargo, estas propuestas están orientadas a interfaces Web tradicionales y no tienen en cuenta la existencia de las RIAs. En referencia a las aplicaciones RIA, Fraternali et al. [16] proponen una extensión para el diseño de RIAs (orientadas al navegador, basadas en AJAX).

A pesar del número de extensiones de WebML, todas ellas permanecen inconexas, es decir, cada una propone unas mejoras en un ámbito que no tienen ninguna conexión, no son aparentemente reutilizadas. Al contrario, S<sup>m</sup>4RIA combina las primitivas para modelar RIAs y aplicaciones Web semánticas en sus modelos que pueden ser aplicadas conjuntamente para desarrollar diferentes tipos de aplicaciones SRIA, entre ellas, nuevos sitios sociales. Asimismo, el método ofrece una mejor separación de características de forma que divide de forma clara los elementos necesarios para modelar la persistencia de datos, la lógica de negocio y la interfaz.

SHDM [17] es un método de diseño de aplicaciones Web semánticas dirigido por ontologías. Años después de su invención fue extendido con las primitivas necesarias para diseñar RIAs, tal y como se describe en [18]. Esta extensión incrementó el número de elementos de la ontología de presentación e incluyó nuevas funcionalidades del generador de sitios Web, p. ej. la generación de etiquetas RDFa.

RUX-Method [19] es una metodología de diseño basado en modelos para el desarrollo de interfaces RIA (únicamente la parte servidora). Como una extensión al método, Linaje et al. [20] presentaron los requisitos y los cambios necesarios para incluir anotaciones semánticas dentro de las IU desarrolladas con AJAX utilizando el estándar de accesibilidad WAI-ARIA del W3C.

Es interesante destacar que los métodos anteriores fundamentan la inclusión de anotaciones semánticas en RIAs en el hecho que estén implementadas con tecnologías basadas en Javascript, p. ej. AJAX. Este hecho tiene una fuerte influencia en el diseño de los métodos y limita evidentemente el tipo de tecnologías que pueden ser utilizadas para generar las RIAs. De forma opuesta, S<sup>m</sup>4RIA provee una solución más general basada en los principios del proyecto Linking Open Data (<http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>) y, consecuentemente, no hay una dependencia fuerte entre el método y la tecnología. Además S<sup>m</sup>4RIA utiliza los modelos de ontología (dominio, navegación y visualización) para generar ontologías que permitan manejar una representación de la SRIA en tres capas que pueda ser “entendida” por los clientes software.

Por último, este análisis no incluirá los métodos de diseño WSDM [21] y Hera [22], ya que a pesar que son capaces de reutilizar conocimiento de la Web y de generar aplicaciones Web semánticas, carecen de primitivas para el diseño de RIAs.

## 7 Conclusiones y trabajos futuros

El presente artículo aplica el concepto de RIA semántica y la metodología S<sup>m</sup>4RIA a la mejora de la interoperabilidad entre sistemas RIA y fuentes de datos externas. El uso de SRIAs como plataforma proporciona un conjunto de componentes que permiten solucionar el problema de interoperabilidad entre sitios sociales. Asimismo, las SRIA ofrecen un conjunto de características ya desarrolladas en el entorno de la Web semántica para favorecer el acceso de agentes software a los datos.

El método S<sup>m</sup>4RIA por su parte facilita a los diseñadores el enriquecimiento de las SRIA con conocimiento desde bases de conocimiento de la Web o la creación de nuevas bases de conocimiento accesibles desde la Web. S<sup>m</sup>4RIA define los modelos y procesos necesarios para representar cómo el conocimiento de la Web es importado y reutilizado en las aplicaciones y qué conocimiento de la aplicación diseñada puede ser compartido utilizando diferentes técnicas. Los modelos ODM permiten la representación de la semántica del dominio de la aplicación y, al mismo tiempo, contienen el conocimiento de diseño de la aplicación, reutilizable en otros diseños.

Los beneficios concretos de utilizar S<sup>m</sup>4RIA son los siguientes: al ser una aproximación basada en modelos 1) permite definir una aplicación Web 3.0 completa en tiempo de diseño y generarla de forma (semi-)automática, lo cual ayuda a reducir el coste asociado de desarrollo y mantenimiento; 2) incluye primitivas para diseñar cómo las aplicaciones reutilizarán y compartirán sus datos en la Web, mejorando la interoperabilidad de los sistemas basados en la plataforma SRIA desde el diseño; y 3) facilita la reutilización de KBs por medio de sus modelos, ayudando a los diseñadores no expertos en el área de la Web Semántica a explotar estos recursos.

Las líneas de trabajo futuro son las siguientes: *a)* aplicar la propuesta en el desarrollo de otros tipos de aplicación, p.ej. blogs o wikis semánticos [7], que utilicen patrones de diseño de Webs sociales; *b)* estudiar técnicas de optimización para recuperar conocimiento desde fuentes externas; y *c)* completar la implementación de S<sup>m</sup>4RIA en OIDE (OOH4RIA Integrated Development Environment, [23]).

**Agradecimientos.** El presente artículo ha sido financiado por el Ministerio de Educación del Gobierno de España bajo el programa de Formación de Profesorado Universitario (FPU, ref. AP2007-03076) y el proyecto SONRIA (TIN2010-15789). Asimismo, los autores agradecen el soporte económico de la Universidad de Alicante mediante el proyecto de investigación DIMENRIA (GRE10-23).

## Referencias

1. Mikroyannidis, A. Towards a Social Semantic Web. *Computer* 40, 11 (Nov. 2007), 113-115. (2007) DOI= <http://dx.doi.org/10.1109/MC.2007.405>
2. O'Reilly, T.: What is web 2.0? Design patterns and business models for the next generation of software. *Inf. Téc.* (2005), <http://oreilly.com/web2/archive/what-is-web-20.html>
3. Breslin, J., Decker, S.: The Future of Social Networks on the Internet. The Need for Semantics. *IEEE Internet Computing* 11(6), 86-90 (2007)

4. Breslin, J.G., Passant, A., Decker, S.: Motivation for Applying Semantic Web Technologies to the Social Web. In *The Social Semantic Web* (2009), eds. J.G. Breslin et al. DOI 10.1007/978-3-642-01172-6\_2
5. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* 284(5), 34–43 (2001), <http://www.sciam.com/article.cfm?id=the-semantic-web>
6. Gruber, T.: Collective Knowledge Systems: Where the Social Web Meets the Semantic Web. *Journal of Web Semantics* (2007)
7. Kinsella, S., Passant, A., Breslin, J.G., Decker, S., Jaokar, A.: The Future of Social Web Sites: Sharing Data and Trusted Applications with Semantics. *Advances in Computers* 76, 121-175 (2009).
8. Hermida, J.M., Meliá, S., Montoyo, A., Gómez, J.: Modelling Semantic Rich Internet Applications Using a Model-Driven Approach. LNCS vol. 6724 (2011), in press.
9. Bizer, C., Heath, T., Berners-Lee, T. Linked Data - The Story So Far. *Int. Journal Semantic Web Information Systems*, 5 (3), 1-22 (2009).
10. Meliá, S., Gómez, J., Pérez, S., Díaz, O.: A model-driven development for GWT-based rich Internet applications with ooh4ria. En *actas del ICWE 2008*, Washington, DC, USA, pp. 13-23, IEEE Computer Society (2008)
11. Object Management Group: *Ontology Definition Metamodel Version 1.0*. (Mayo 2009), <http://www.omg.org/spec/ODM/1.0/PDF> (Revisado: 1 julio 2011)
12. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): a modeling language for designing Web sites. *Computer Networks* 33(1–6), 137–157 (2000)
13. Ceri, S., Brambilla, M., Fraternali, P.: The history of WebML. In *Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos*, pp. 273–292, Springer, Heidelberg (2009)
14. Brambilla, M., Facca, F.M.: Building semantic web portals with WebML. LNCS vol. 4607, pp. 312–327, Springer, Heidelberg (2007)
15. Brambilla, M., Ceri, S., Facca, F.M., Celino, I., Cerizza, D., Valle, E.D.: Model-driven design and development of semantic web service applications. *ACM Trans. Internet Technol.* 8(1), 3 (2007)
16. Fraternali, P., Comai, S., Bozzon, A., Carughi, G.T.: Engineering rich internet applications with a model-driven approach. *ACM Trans. Web* 4(2), 1–47 (2010)
17. Lima, F., Schwabe, D.: Modeling applications for the semantic web. LNCS 2722 pp. 417–426, Springer, Heidelberg (2003)
18. SHDM - Semantic Hypermedia Design Method. [http://www.w3.org/2005/Incubator/model-based-ui/wiki/SHDM\\_-\\_Semantic\\_Hypermedia\\_Design\\_Method](http://www.w3.org/2005/Incubator/model-based-ui/wiki/SHDM_-_Semantic_Hypermedia_Design_Method)
19. Preciado, J.C., Linaje, M., Comai, S., Sanchez-Figueroa, F.: Designing rich internet applications with web engineering methodologies. En *actas del WWV 2007*, Washington DC, USA, pp. 23–30, IEEE Computer Society (2007)
20. Linaje, M., Lozano-Tello, A., Preciado, J.C., Rodríguez, R., Sanchez-Figueroa, F.: Obtaining accessible RIA UIs by combining RUX-Method and SAW. En *actas del WWV 2009*, Linz, Austria, pp. 85-97 (2009)
21. De Troyer, O., Casteleyn, S., Plessers, P.: WSDM: Web Semantics Design Method. In *Web Engineering: Modelling and Implementing Web Applications*, pp. 303-351, Human-Computer Interaction Series, Springer, London (2007)
22. Van der Sluijs, K., Houben, G., Broekstra, J., Casteleyn, S.: Hera-S: web design using sesame. En *actas del ICWE 2006*. Palo Alto, California, USA, pp. 337-344 (2006)
23. Meliá, S., Martínez, J., Mira, S., Osuna, J., Gómez, J.: An Eclipse Plug-in for Model-Driven Development of Rich Internet Applications. LCNS 6189, pp. 514-517 (2010)