

# AVATAR: Un sistema de recomendación personalizada de contenidos televisivos basado en información semántica\*

Manuel Ramos Cabrer, Yolanda Blanco Fernández, Alberto Gil Solla  
Departamento de Ingeniería Telemática. Universidad de Vigo  
ETSI de Telecomunicación. C/ Maxwell s/n. Campus Universitario.  
36310 - Vigo (Pontevedra)  
Teléfono: 986 81 38 70 Fax: 986 81 21 16  
E-mail: mramos,yolanda,agil@det.uvigo.es

**Abstract** *In this paper we present a recommender system of personalized TV contents, named AVATAR, for which we propose a multi-agent architecture, that combines different knowledge inference strategies. We focus on the description of one of these strategies, the semantic one, explaining an example in the context of personalized digital television. Our goal is to take advantage of the experience of the Internet, from the current syntactic search engines to the newest results in Semantic Web area. Trying to be always under the umbrella of the most used standards, we adopt the TV-Anytime specification for metadata definition of TV contents, the OWL language to build a suitable ontology for knowledge representation, and the MHP specification for system development.*

## 1. Introducción

En la actualidad estamos asistiendo a un cambio fundamental en la TV: la migración de la TV analógica a la digital. Este cambio tiene dos implicaciones principales: un considerable aumento en la capacidad para difundir más canales en el mismo ancho de banda, y la posibilidad de enviar aplicaciones software conjuntamente con los contenidos audiovisuales, posibilitando la aparición de nuevas oportunidades de negocio [7].

En este nuevo escenario, los usuarios podrán acceder a un mayor número de canales de distintos proveedores y a muchos más contenidos que hoy en día. Por ello, serán necesarias nuevas herramientas que ayuden a los usuarios en la selección de contenidos que pueden ser de su interés. En el campo de la TV, los sistemas recomendadores pueden permitir al usuario jugar un papel activo y buscar contenidos concretos [9] o, por el contrario, pueden analizar las preferencias de los usuarios, así como los programas que éstos han visto, con el fin de recomendar contenidos personalizados sin una solicitud explícita previa. Esta última posibilidad pretende reducir la implicación de los usuarios en procesos de búsqueda tediosos. En este artículo, presentamos un sistema recomendador, denominado AVATAR, que explora ambas posibilidades.

A medida que la implantación de la TV digital va avanzando, los recomendadores personalizados van ganando cada vez más interés [8], dando como resultado la aparición de diversas aproximaciones, como pueden ser: sistemas expertos, técnicas bayesianas, métodos basados en contenidos, filtrado colaborativo, árboles de decisión o redes neuronales. Con el fin de conseguir las recomendaciones más adecuadas para el usuario, el sistema recomendador que proponemos combina algunas de las técnicas anteriores con un razonamiento semántico sobre los contenidos televisivos, los perfiles de los usuarios y los registros históricos de visionado de contenidos. Estos mecanismos semánticos necesitan, sin duda, un alto grado de normalización y formalización. En este sentido, debemos indicar que utilizamos el estándar TV-Anytime, que normaliza las descripciones de contenidos genéricos de TV, de instancias concretas de programas y de perfiles de usuario. Constituye,

por tanto, un marco adecuado para referenciar, localizar y procesar contenidos.

A semejanza de lo ocurrido con Internet, es de esperar la aparición de un nuevo escenario en el que estén disponibles diversos sistemas recomendadores, compitiendo por ser el preferido de los usuarios. Por tanto, no parece adecuado que el software recomendador se instale durante la fabricación del receptor de TV (Set-Top Box o STB). Por el contrario, proponemos la utilización de aplicaciones descargadas desde el proveedor de servicio a través del flujo de transporte. En el campo de la TV digital en Europa, la adopción del estándar MHP [4] previsiblemente solucionará los problemas actuales de normalización de los equipos receptores, dando lugar a un mercado horizontal para los proveedores de contenidos, proveedores de servicios, plataformas digitales, operadores de red y usuarios. En este modelo de mercado, parece más adecuado el desarrollo de una aplicación independiente de la plataforma que permita realizar recomendaciones no ligadas a intereses comerciales concretos. Es previsible considerar que este modelo requerirá que el coste de las recomendaciones realizadas sea asumido por el usuario, bien directamente o bien a través de publicidad.

Este artículo se organiza de la siguiente forma: la siguiente sección describe la arquitectura del sistema que proponemos. En la sección 3 presentamos el marco conceptual definido para dar soporte al recomendador semántico propuesto, el cual se presenta en la sección 4. La sección 5 presenta un ejemplo del tipo de razonamientos soportados por el recomendador. Finalmente, presentamos algunas conclusiones y líneas de trabajo futuro.

## 2. Arquitectura del sistema AVATAR

En esta sección presentamos las principales decisiones de diseño de la arquitectura del recomendador AVATAR. El objetivo es que el sistema presente un alto grado de modularidad con el fin de que se puedan añadir fácilmente nuevas funcionalidades relacionadas con la elaboración de recomendaciones personalizadas.

A la hora de diseñar el sistema, nos ha preocupado especialmente el desarrollo eficiente de distintas estrategias

\*Este trabajo ha sido subvencionado por el Ministerio de Educación y Ciencia dentro del proyecto TSI2004-03677

para elaborar sugerencias de contenidos televisivos ya que las técnicas actuales tienen una capacidad de razonamiento bastante limitada de forma individual. Por tanto, proponemos una arquitectura abierta que: (i) permita actualizar los módulos que generan recomendaciones y añadir nuevos módulos que calculen sugerencias siguiendo diferentes estrategias, y (ii) soporte mecanismos para combinar varias recomendaciones. En concreto, en la versión actual del sistema, proponemos una aproximación que combina un modelo bayesiano y técnicas de razonamiento semántico, una metodología de inferencia usual en el campo de la Web Semántica [5, 3], pero novedosa en el campo de los recomendadores televisivos.

La información sobre los usuarios que necesita el recomendador incluye, al menos, datos personales objetivos, sus preferencias en relación con los contenidos televisivos y registros históricos que almacenen los programas vistos en el pasado. Además de los datos proporcionados por los usuarios, AVATAR debe tener en cuenta el éxito obtenido en recomendaciones previas con el fin de mejorar la calidad de las sugerencias futuras. Esta información de realimentación permite conseguir una rápida convergencia entre las preferencias del usuario y las recomendaciones realizadas.

## 2.1. Una aplicación interactiva MHP

Como comentamos en la introducción, el sistema debe ser lo suficientemente flexible para actualizarse con frecuencia. Por este motivo, proponemos utilizar aplicaciones interactivas descargadas desde el proveedor de servicio hasta el STB a través del flujo de transporte. Actualmente existen importantes esfuerzos para normalizar este tipo de aplicaciones. Distintas organizaciones como DBV ([www.dvb.org](http://www.dvb.org)) y ATSC ([www.atsc.org](http://www.atsc.org)) han identificado la necesidad de un proceso de normalización para asegurar la interoperabilidad entre aplicaciones de diferentes proveedores y sobre STB de diferentes fabricantes. Esta normalización se ocupa del ciclo de vida de las aplicaciones, los mecanismos de control y del API disponible localmente para las aplicaciones, de forma independiente de los recursos de un receptor concreto.

La iniciativa de normalización que más ha avanzado es DVB MHP, estándar que sigue nuestra propuesta. En este estándar, las aplicaciones se ejecutan ligadas a servicios concretos o eventos (programas de TV) en un servicio, y, normalmente, no sobreviven a la finalización de ese contexto (*cambio de servicio o final de evento*). No obstante, en este tipo de recomendadores es necesario un proceso de realimentación persistente para medir el éxito de las sugerencias, almacenando referencias a los programas vistos por los usuarios para analizar el resultado de las recomendaciones realizadas. Esta tarea se extiende más allá de un servicio o programa concreto, por lo que no puede ser realizada por una aplicación MHP. Por tanto, surge la necesidad de un agente especial que almacene de manera ininterrumpida las acciones del usuario. En nuestra propuesta este agente, denominado **agente local** se integra en el software del sistema.

Otro aspecto que necesita normalización es el referente a los formatos de la información que describe la programación de los canales de televisión. Actualmente las diferentes plataformas digitales envían una pequeña cantidad de datos sobre los eventos que van a programar, con el fin de permitir la elaboración de Guías Electrónicas de Progra-

mación que muestren los contenidos de TV de una manera organizada. Esta información es insuficiente para procesos complejos, como pueden ser aquellos realizados por recomendadores basados en razonamiento semántico, y, por tanto, se debe ampliar.

Si esta información es enviada por el proveedor de servicios, podría darse el caso de que tenga un formato de almacenamiento propietario, algo no deseable en un escenario en el que estén compitiendo varios servicios de recomendación, ya que podría suponer la existencia de gran cantidad de información redundante sobre la programación. Lo deseable será que esta información se envíe una sola vez en un formato normalizado y esté disponible para todos los proveedores de servicio. Hoy en día existe una iniciativa muy importante promovida por el Foro TV-Anytime ([www.tv-anytime.org](http://www.tv-anytime.org)), que pretende normalizar los formatos de los metadatos que describan los contenidos de TV y los perfiles de los usuarios. Las principales organizaciones de normalización (DVB y ATSC) están trabajando en integrar TV-Anytime en sus respectivas infraestructuras.

En nuestro prototipo hemos utilizado los formatos definidos por TV-Anytime para almacenar los registros históricos de los usuarios y sus preferencias personales sobre contenidos televisivos. Estos datos se almacenan en los perfiles de usuario, junto con información adicional, como comentaremos en el apartado 3.2. Toda esta información deberá ser almacenada de manera permanente en el propio STB por parte del agente local. Este hecho revela una necesidad adicional de estandarización: el acceso a esta información debería realizarse de forma normalizada ya que el formato concreto de almacenamiento será desconocido. Este aspecto aún no ha sido abordado, ni por MHP ni por TV-Anytime. Por esta razón, hemos introducido una nueva API MHP – la API MHP TV-Anytime – cuyo propósito es proporcionar una forma independiente y neutral de acceder a la información descrita mediante los metadatos TV-Anytime, aún cuando el agente local no utilice el formato TV-Anytime para almacenar estos datos.

## 2.2. AVATAR: un sistema multi-agente

Considerando todos los aspectos indicados, proponemos un arquitectura modular y abierta (Fig. 1), en la que el recomendador está dividido en dos partes. La primera está compuesta por el software local del STB que, por una parte solicita los datos personales y las preferencias de los usuarios, y por otra almacena la información relativa a los contenidos de TV ya vistos. Esta información estará accesible al recomendador a través del API MHP TV-Anytime (en formato TV-Anytime) La segunda parte consiste en la aplicación MHP que implementa el servicio de recomendación, y en la que identificamos tres módulos:

**Recomendadores.** Son varios agentes que implementan las diferentes estrategias para realizar las recomendaciones personalizadas. La figura 1 muestra los tres implementados en la versión actual de nuestro recomendador: un agente basado en técnicas bayesianas, uno basado en razonamiento semántico y otro basado en coincidencia de perfiles<sup>1</sup>. Cada uno de estos agentes realiza recomendaciones de forma independiente (B-REC, S-REC y P-REC) y el módulo de combinación toma como entrada estas tres sugerencias para calcular una recomendación global (G-REC) El agente de combinación es una red neuronal, cuya utilización

<sup>1</sup>En este artículo nos centraremos en la descripción del agente basado en razonamiento semántico. El lector interesado en el diseño de los otros agentes puede consultar [1].

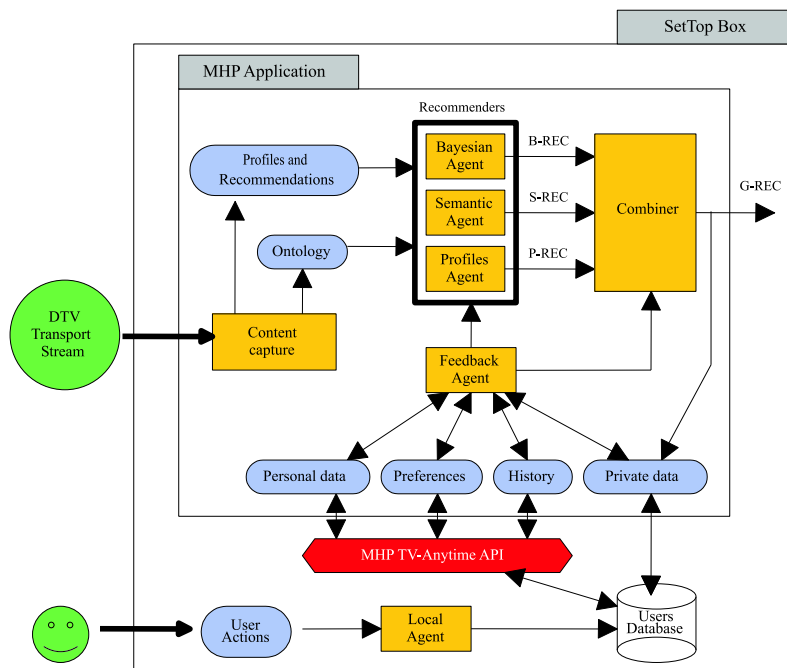


Figura 1: Arquitectura del sistema recomendador

como estrategia de combinación está justificada en diversos trabajos, como puede ser [11]. La recomendación final se almacena en el STB y se compara con las decisiones del usuario para mejorar futuras sugerencias.

**Recogida y clasificación de la información.** Los algoritmos utilizados por AVATAR necesitan información sobre la programación de los canales de TV y un grupo de perfiles de usuario prototípicos con recomendaciones generales para cada uno de estos perfiles. Este subsistema se encarga de extraer la información comentada de los flujos de transporte y la clasifica adecuadamente para su almacenamiento. En nuestra propuesta, toda la información sobre contenidos de TV se recibe en formato TV-Anytime.

**Realimentación.** Como ya hemos comentado, la información con la que trabaja el sistema recomendador debería actualizarse a menudo para que las preferencias asociadas a los perfiles de los usuarios sean adecuadas. Para llevar a cabo este proceso, AVATAR necesita acceder a la información proporcionada por los espectadores una vez se les hayan presentado las sugerencias. Para ello el agente de realimentación accede a la información almacenada en la base de datos que gestiona el agente local a través de la API MHP TV-Anytime, y actualiza los perfiles de usuario. Además, el agente de realimentación realimenta el módulo formado por los agentes recomendadores con la información que éstos necesitan, y que dependerá de los métodos que implementen. Así, los agentes semánticos necesitan información actualizada sobre las preferencias de los usuarios y el conocimiento almacenado en la ontología de TV implementada. Los agentes basados en coincidencia de perfiles utilizan perfiles prototípicos recibidos desde el proveedor e información sobre los gustos de los usuarios sobre contenidos de TV. En lo referente a los agentes bayesianos, es necesaria información referente a los programas ya vistos y sobre las preferencias de los usuarios registrados en el sistema.

### 3. El marco de razonamiento

En esta sección describimos los elementos que integran el marco de razonamiento propuesto en nuestro enfoque. Empezaremos presentando la base de conocimiento del sistema, es decir, la ontología que hemos implementado. A continuación, propondremos un forma de representación de las preferencias de los usuarios basada en la reutilización de dicha ontología. Finalmente, presentamos una clasificación de las diferentes relaciones semánticas que nuestro recomendador puede inferir.

#### 3.1. La ontología televisiva

Para obtener un recomendador de contenidos personalizados basado en razonamiento semántico es necesario representar de forma organizada el conocimiento del dominio de aplicación concreto del sistema, en este caso, la TV. En el marco de la Web Semántica una de las tecnologías de uso más extendido para este tipo de propósitos son las ontologías [10]. Por esta razón, hemos optado por implementar, mediante el lenguaje OWL [6] y utilizando la herramienta Protégé-2000 (<http://protege.stanford.edu>), una ontología sobre el dominio de la TV<sup>2</sup>.

Así, hemos definido las diferentes entidades relacionadas con los contenidos audiovisuales que queremos considerar en nuestro análisis, recurriendo para ello a un conjunto de clases. Por ejemplo, en la ontología definimos una clase raíz, *TV Contents*, de la que dependerán las diferentes categorías de programas considerados en nuestro enfoque (*Movies*, *TV Series*, *Informative Programs*, *Sport Programs*, etc.). Del mismo modo, también definiremos clases para identificar a las personas que participan en los programas (*Presenters*, *Directors*, *Actors*, *Actresses*, *Producers*, *ScreenWriters*, etc.)

Las especializaciones relativas a las clases descritas antes, son traducidas a la ontología en forma de subclases (e.g. *Action Movies*, *Terror Series*, *National News*, *Soccer*

<sup>2</sup>Esta ontología está disponible en <http://avatar.det.uvigo.es/ontology>.

Broadcasting, Movie Directors, StarringActors, etc.)

Las características de los diferentes programas de TV que define la especificación de metadatos de TV-Anytime, han sido introducidos en nuestra base de conocimiento mediante propiedades, también organizadas jerárquicamente. Hemos utilizado las diferentes propiedades que define OWL – funcionales, inversas, simétricas, transitivas, entre otras –, con el objeto de enriquecer la base de conocimiento, y facilitar los procesos de inferencia.

Una vez definida la jerarquía de clases y propiedades de la ontología OWL, se hace necesario instanciar esta base de conocimiento, utilizando para ello programas de TV concretos. En este proceso, se ha analizado la variada oferta de las principales plataformas de TV disponibles, con el fin de obtener una ontología completa a partir de la cual, validar el enfoque de razonamiento propuesto.

### 3.2. Los perfiles de usuario

Los perfiles de usuario, son uno de los elementos clave en el proceso de personalización, ya que es necesario un mecanismo para representar formalmente el conocimiento relativo a las preferencias de los usuarios. Cada perfil definido en AVATAR almacena información personal sobre el usuario, así como sus preferencias. Para modelar estas preferencias hemos usado un subconjunto dinámico de la ontología OWL de TV descrita antes, construido de forma incremental a medida que el sistema conoce información adicional sobre el espectador. Dicho subconjunto, integrado por propiedades, clases e instancias concretas, recibe el nombre de *contexto semántico personalizado* ó *CS<sub>P</sub>*. Por razones obvias, los perfiles de usuario en AVATAR reciben el nombre de *perfiles-ontología*. Concretamente, cuando el usuario ve un nuevo contenido, AVATAR añade a su perfil (i) esa instancia, (ii) la clase a la que ésta pertenece, (iii) la jerarquía completa de superclases relativas a ella en la ontología de TV y (iv) algunas propiedades que permiten identificar características de los programas relevantes en nuestro entorno de personalización (género, créditos del programa, palabras claves, etc.).

Merece la pena destacar que los *perfiles-ontología* superan claramente el empleo de listas planas contemplado en enfoques anteriores. En éstas, se identifican las preferencias de los usuarios pero no se estructuran de forma que se favorezca el descubrimiento de nuevo conocimiento, objetivo claramente perseguido en nuestro recomendador semántico.

Por otra parte, para mantener preferencias de usuario permanentemente actualizadas, nuestra propuesta define tres índices que estarán asociados a cada instancia y a cada clase contenida en el *perfil-ontología* del usuario. Estos índices son actualizados por el agente de realimentación de forma automática a partir de las acciones llevadas a cabo por los espectadores (por ejemplo, qué programas son aceptados y cuáles son rechazados de entre todos los sugeridos por el recomendador), tal como se describe en [2]. Los índices considerados son:

- **Nivel de Interés** ó DOI: Cuantifica el interés o desinterés del usuario en la instancia/clase a la que dicho índice esté asociado. Los valores positivos están reservados para aquellas instancias del *perfil-ontología* que interesan al usuario y a las que hemos denominado preferencias positivas. Por el contrario,

las preferencias negativas hacen referencia a aquellas instancias que no agradan al espectador.

- **Confianza**: Este índice se usa para cuantificar el éxito o fracaso del recomendador en las sugerencias ofrecidas en el pasado.
- **Relevancia**: Resulta de la combinación de los dos primeros y es usado por el recomendador para ordenar los contenidos finalmente sugeridos al usuario.

El proceso de razonamiento que proponemos partirá de la información expresada en el *CS<sub>P</sub>* del usuario y, a partir de ella, calculará las recomendaciones oportunas.

### 3.3. Clasificación de relaciones semánticas

Una vez que disponemos de la información de partida del proceso de razonamiento en los perfiles de usuario, y antes de describir este proceso, es necesario presentar las diferentes relaciones semánticas que AVATAR puede inferir a partir de dicha información.

En realidad, el conjunto de información de partida del proceso de razonamiento no será sólo el formado por las clases y propiedades definidas en el *CS<sub>P</sub>* del espectador, sino que estará formado por esas clases y propiedades junto con aquellas relacionadas con ellas desde un punto de vista semántico. Definimos esta relación mediante los siguientes conceptos:

- **Nexo semántico entre propiedades**: Dadas dos propiedades  $p_i$  y  $p_j$ , decimos que existe un nexo semántico entre ellas (lo que expresaremos como  $p_i \leftrightarrow p_j$ ) si se cumple alguna de las siguientes condiciones:

- $p_i = p_j$
- $p_i$  es la superpropiedad directa de  $p_j$  en la jerarquía definida en la ontología<sup>3</sup>. Expresaremos esta condición como  $p_j \sqsubseteq_D p_i$ .
- $p_j$  es la superpropiedad directa de  $p_i$  en la jerarquía.
- $\exists q$ , tal que  $p_i, p_j \sqsubseteq_D q$ : Es decir,  $p_i$  y  $p_j$  son propiedades hermanas.
- $\exists r$  y  $q$ , tales que  $p_i \sqsubseteq_D r \sqsubseteq_D q$  y  $p_j \sqsubseteq_D q$ . Es decir,  $p_j$  es propiedad hermana de la superpropiedad directa de  $p_i$ .
- $\exists r$  y  $q$ , tales que  $p_j \sqsubseteq_D r \sqsubseteq_D q$  y  $p_i \sqsubseteq_D q$ . Es decir,  $p_j$  es subpropiedad directa de una propiedad hermana de  $p_i$ .
- $p_i$  es propiedad inversa de  $p_j$  (representaremos la inversa de  $p_j$  como  $\neg p_j$ )

En la notación introducida antes, esta definición sería:

$$\begin{aligned}
 p_i \leftrightarrow p_j &\Leftrightarrow p_i = p_j \vee \\
 &p_j \sqsubseteq_D p_i \vee \\
 &p_i \sqsubseteq_D p_j \vee \\
 &\exists q / p_i, p_j \sqsubseteq_D q \vee \\
 &\exists r, q / p_i \sqsubseteq_D r \sqsubseteq_D q \wedge p_j \sqsubseteq_D q \vee \\
 &\exists r, q / p_j \sqsubseteq_D r \sqsubseteq_D q \wedge p_i \sqsubseteq_D q \vee \\
 &p_i = \neg p_j
 \end{aligned}$$

<sup>3</sup>De todas las propiedades relacionadas mediante *subsumption* con  $p_i$  (representado como *Properties*  $\sqsubseteq p_i$ ), únicamente consideraremos la(s) superpropiedad(es) directa(s). Para identificar esta relación utilizaremos el operador  $\sqsubseteq_D$ .

- **Nexo semántico entre clases:** Análogamente, definimos un nexo semántico entre las clases  $C_i$  y  $C_j$  como:

$$\begin{aligned} C_i \leftrightarrow C_j \Leftrightarrow & C_i = C_j \vee \\ & C_j \sqsubseteq_D C_i \vee \\ & C_i \sqsubseteq_D C_j \vee \\ & \exists Q^* \neq \text{Ontology Root} / C_i, C_j \sqsubseteq_D Q^* \vee \\ & \exists A, B / C_i \sqsubseteq A \sqsubseteq_D B \wedge C_j \sqsubseteq B \vee \\ & \exists A, B / C_j \sqsubseteq_D A \sqsubseteq_D B \wedge C_i \sqsubseteq_D B \end{aligned}$$

En la definición anterior,  $Q^*$  debe ser necesariamente diferente del elemento raíz de la ontología. De lo contrario, estaríamos relacionando contenidos muy diferentes entre sí, solamente por ser subclases directas del elemento raíz.

Una vez que hemos establecido los criterios para descubrir relaciones entre clases y propiedades, nos centramos en las asociaciones establecidas entre las diferentes instancias declaradas en la ontología del sistema.

Empecemos considerando que dos instancias de clases definidas en una ontología pueden estar relacionadas **explícitamente** a través de propiedades *Object* declaradas en la base de conocimiento, o bien puede haber **relaciones implícitas** entre ellas, a través de instancias de otras clases. Para referirnos a relaciones semánticas *explícitas*, emplearemos la notación  $C_1 [p] C_2$ , siendo  $p$  una propiedad *Object*, y  $C_1$  y  $C_2$  las clases a las que pertenecen las instancias entre las que se establece la relación.

El descubrimiento de relaciones *implícitas* será el objetivo del recomendador **AVATAR** durante el proceso de razonamiento semántico, ya que permitirá obtener mejores recomendaciones con un alto grado de personalización. En concreto, en nuestra propuesta definimos los siguientes tipos de relaciones semánticas *implícitas*:

- **Cadena semántica directa** ( $\Rightarrow$ ): Está formado por varias relaciones semánticas *explícitas* de la forma:

$$D_0 [p_0] D_1 [p_1] D_2 \dots D_{q-1} [p_{q-1}] D_q$$

Este tipo de relación se basa en el principio de transitividad (si  $a \rightarrow b$  y  $b \rightarrow c$  entonces  $a \rightarrow c$ ), y permite una relación semántica entre instancias de las clases  $D_0$  y  $D_q$ . Lo denotaremos como  $D_0 \Rightarrow D_q$ .

- **Cadenas semánticas directas con nodos comunes** ( $\Rightarrow_{\text{Nodos}}$ ): Consideremos dos cadenas semánticas directas, tales que en ambos participan instancias pertenecientes a una misma clase (en adelante, a este conjunto de clases comunes les llamaremos *Nodos*). Por ejemplo,

$$DSC_1 = E_0 [p_0] E_1 [p_1] \mathbf{N} \dots \mathbf{M} [p_{m-1}] E_m$$

$$DSC_2 = D_0 [q_0] \mathbf{M} [q_1] D_2 \dots \mathbf{N} [q_{n-1}] D_n$$

$$DSC_1 \cap DSC_2 = \{\text{Nodos}\} = \{\mathbf{M}, \mathbf{N}\}$$

El sistema puede inferir las relaciones semánticas  $E_0 \Rightarrow E_m$  y  $D_0 \Rightarrow D_n$  debido a la existencia de dos *cadena semánticas directas*. Además por el hecho de haber nodos comunes entre ambas cadenas ( $M$  y  $N$ ), también se puede establecer una relación semántica entre  $E_0$  y  $D_0$ . Esta relación es de tipo *cadena semánticas directas con nodos comunes*, que denotaremos como  $E_0 \Rightarrow_{M-N} D_0$ .

- **Cadenas semánticas directas mixtas** ( $\Rightarrow_{\text{Nodos}}$ ): Es una combinación de las dos relaciones semánticas anteriores. Así, si una instancia de la clase  $C_1$  está rela-

cionada con otra de la clase  $C_2$  mediante una cadena semántica directa con nodos comunes ( $C_1 \Rightarrow_{\text{Nodos}} C_2$ ), y a la vez  $C_2$  está relacionada con otra instancia de  $C_3$  mediante una cadena semántica directa ( $C_2 \Rightarrow C_3$ ), podemos establecer una nueva relación entre  $C_1$  y  $C_3$  gracias al principio de transitividad. A este tipo de asociación semántica la denominaremos *cadena semántica directa mixta*, que denotaremos como  $C_1 \Rightarrow_{\text{Nodos}} C_3$ .

- **Cadenas semánticas indirectas** ( $\approx$ ): Dadas dos *cadena semánticas directas*  $DSC_1$  y  $DSC_2$ , de longitudes  $l_1$  y  $l_2$ , respectivamente, con  $l_1 \leq l_2$ :

$$DSC_1 = E_0 [p_0] E_1 [p_1] E_2 \dots [p_{l_1-2}] E_{l_1-1}$$

$$DSC_2 = F_0 [q_0] F_1 [q_1] F_2 \dots [q_{l_2-2}] F_{l_2-1}$$

Decimos que existe una *cadena semántica indirecta* entre las instancias de las clases  $E_0$  y  $F_0$ , si las propiedades involucradas en  $DSC_1$  y  $DSC_2$ , están relacionadas una a una mediante *nexo semántico*. Si las longitudes de ambas cadenas puede son diferentes, la condición debe verificarse hasta que se termine la cadena más corta de ambas. Es decir, para las dos *cadena semánticas directas* presentadas arriba, se cumplirá  $E_0 \approx F_0 \Leftrightarrow p_i \leftrightarrow q_i \forall i (0 \leq i \leq l_1 - 2)$

- **Cadenas semánticas virtuales** ( $\approx_{\Rightarrow}$ ,  $\approx_{\Rightarrow_{\text{Nodos}}}$ ): Las *cadena semánticas indirectas* permiten inferir otro tipo de asociaciones en la base de conocimiento del sistema. Estas nuevas asociaciones se pueden clasificar en dos tipos. El primero, que denominaremos *cadena semántica virtual de tipo I*, se obtendrá combinando *cadena semánticas directas* y *cadena semánticas indirectas* mediante el principio de transitividad (así por ejemplo, si  $C_1 \approx D_1$  y  $D_1 \Rightarrow D_3$ , entonces  $C_1 \approx_{\Rightarrow} D_3$ ). De forma similar, podemos obtener *cadena semánticas virtuales de tipo II* combinando *cadena semánticas directas con nodos comunes* y *cadena semánticas indirectas*. Los operadores de ambos tipos de *cadena virtuales* son  $\approx_{\Rightarrow}$  y  $\approx_{\Rightarrow_{\text{Nodos}}}$  respectivamente.

En la figura 2 se muestran algunos ejemplos de las relaciones semánticas definidas.

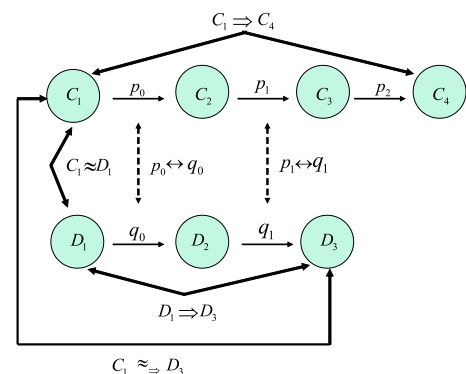


Figura 2: Relaciones semánticas inferidas

## 4. Inferencia de relaciones semánticas

En esta sección presentamos el proceso que llevará a cabo el agente recomendador semántico, dentro del mar-

co presentado antes, para proporcionar una recomendación adecuada y personalizada a un usuario.

El algoritmo que ejecutará este agente (Algoritmo 1) recibe como entrada el  $\mathcal{CSP}$  del usuario y devuelve una relación priorizada de programas como recomendación final del sistema al usuario.

En primer lugar, será necesario calcular las clases y propiedades de interés en el proceso de razonamiento semántico. Para ello el algoritmo recorre las clases y propiedades definidas en el  $\mathcal{CSP}$  del usuario, y calcula los conjuntos de *clases objetivo* ( $TC_{C_i}$ ) (que contendrá las clases que estén relacionadas mediante *nexo semántico* con la clase  $C_i$  definida en el  $\mathcal{CSP}$  del usuario) y de *propiedades objetivo* ( $TP_{C_i}$ ) (que contendrá las propiedades relacionadas con el conjunto de propiedades asociado a la clase  $C_i$  en el  $\mathcal{CSP}$  del usuario) A continuación, es necesario recorrer tanto las instancias definidas en cada una de las clases del  $\mathcal{CSP}$  del usuario, como las instancias de las clases definidas en el conjunto  $TC_{C_i}$  calculado (en el código identificadas como *SetOfInstances*). Con cada una de dichas instancias ( $Prog_{C_j}[k]$  en el código) se invoca la función *Direct\_Semantic\_Chains*, para calcular todas las cadenas semánticas directas relevantes. A partir de este conjunto de cadenas semánticas se pueden descubrir el resto de asociaciones semánticas. Para ello se invocan las funciones correspondientes con los parámetros adecuados.

Una vez descubiertas las asociaciones semánticas, será necesario elaborar la recomendación personalizada. Para ello, el sistema utiliza la función *Recommend*. Esta función clasifica las relaciones descubiertas en base a una estimación de su relevancia. Así por ejemplo, asigna un peso más alto a aquellas relaciones en que participan valores concretos y clases incluidas en el perfil del usuario.

#### Algoritmo 1

```

do_Semantic_Recommendation( $\mathcal{CSP}$ ) {
  for each class  $C_j \in \mathcal{CSP}$  {
     $TC_{C_j} = Target\_Classes(C_j)$ 
     $TP_{C_j} = Target\_Properties(R_{C_j})$ 
    for ( $k = 0$  to  $|SetOfInstances|$ )
       $DSC = Direct\_Semantic\_Chains(Prog_{C_j}[k])$ 
  }
  for ( $j = 0$  to  $|DSC|$ ) {
    for ( $k = j+1$  to  $|DSC|$ ) {
       $Rel = Direct\_Semantic\_Chains\_Common\_Nodes(DSC_j, DSC_k)$ 
       $DSCCN = DSCCN \sqcup Rel$ 
       $ISC = ISC \sqcup Indirect\_Semantic\_Chains(DSC_j, DSC_k)$ 
    }
  }
   $MDSC = Mixed\_Direct\_Semantic\_Chains(DSC, DSCCN)$ 
   $VSC = Virtual\_Semantic\_Chains(DSC, DSCCN, ISC)$ 
   $Recommend(DSC, DSCCN, MDSC, ISC, VSC)$ 
}

```

## 5. Un ejemplo de recomendación

Asumamos un *perfil-ontología* que contiene los siguientes programas: (i) una retransmisión de Formula 1 donde interviene el piloto Fernando Alonso, y (ii) una película de acción, cuya trama discurre en Mexico, protagonizada por

Clint Eastwood. A partir de esta descripción es fácil ver que las clases del  $\mathcal{CSP}$  de este perfil constará de las siguientes clases y propiedades, representadas mediante  $C_i$  y  $R_{C_i}$  ( $i=0,1$ ), respectivamente:

$C_0 = Formula\ 1\ Broadcasting$   
 $R_{C_0} = \{hasDrivers\}$   
 $C_1 = Action\ Movies$   
 $R_{C_1} = \{hasPlace, hasStarringActor, hasName\}$

Supongamos que nuestra base de datos contiene, entre otros, los siguientes programas de TV como instancias de clases de la ontología de TV:

- Una comedia (clase *Comedy Movies*) en la que “Clint Eastwood”, un actor que puede interesar al usuario, participando como actor secundario.
- Un concurso de TV (clase *Quiz Shows*) presentado por una actriz de telenovelas (clase *Soap Operas*), que ha protagonizado la comedia anterior.
- Una película basada en los asesinatos de hinchas de un equipo de fútbol (instancia de la clase *Terror Movies*) cuya trama transcurre en el estadio de fútbol “Amsterdam Arena”, producida por “Clint Eastwood”.
- Un partido de fútbol del mundial 2002 (clase *Soccer Broadcastings*), en el que jugaban, entre otros, el futbolista “Michael Owen”, celebrado en el estadio “Yokohama”.
- Un documental deportivo (clase *Sport Documentaries*) sobre los deportistas de élite, con participaciones del piloto “Fernando Alonso” y del futbolista “Beckham”.

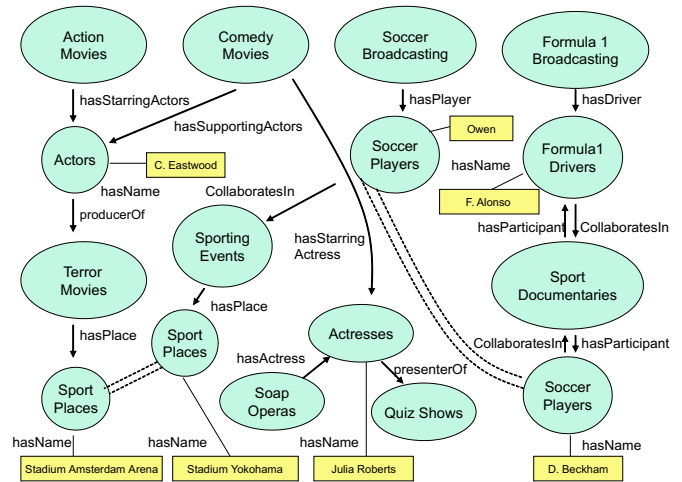


Figura 3: Subconjunto de las instancias de la ontología

En la figura 3 se muestra una parte de la ontología de TV implementada<sup>4</sup>, que contiene los programas que acabamos de definir. En realidad, las clases del ejemplo tienen muchas más propiedades de las mostradas, pero el subconjunto mostrado es suficiente para ilustrar el ejemplo que nos ocupa. Esto también nos permite demostrar que el sistema de recomendación es capaz de inferir relaciones semánticas en la base de conocimiento, aun cuando la información proporcionada sea incompleta o escasa. Como se aprecia en la figura, emplearemos rectángulos para definir los valores tomados por las propiedades *Datatype*.

Veamos en qué medida son importantes los programas que hemos representado en la figura 3, de cara a una recomendación personalizada. Tal como se ha descrito en el

<sup>4</sup>En esta figura, mostramos únicamente un subconjunto reducido de instancias. La jerarquía de clases se ha omitido por falta de espacio.

algoritmo *do\_Semantic\_Recommendation*, el primer paso será calcular el conjunto de clases y propiedades de interés a la hora de razonar (*clases objetivo y propiedades objetivo*), a partir de la información del *CSP* del usuario. Así, la función *Target\_Classes*, permite obtener las clases relevantes para el proceso de recomendación, asociadas a las clases  $C_0$  y  $C_1$  del *CSP*. En nuestro ejemplo, éstas serían:

- $TC_{C_0}$ , formado por *Formula 1 Broadcasting* y *Soccer Broadcasting* (clase hermana de  $C_0$ )
- $TC_{C_1}$ , formado por *Action Movies* y *Soap Operas* (hermana de la superclase de  $C_1$ )

En el ejemplo mostrado en la figura 3, todas las propiedades definidas son relevantes en el proceso de razonamiento semántico (por pertenecer a  $TP_{C_0}$  ó  $TP_{C_1}$ )

Una vez calculados los conjuntos de clases y propiedades en las que se basará el sistema para razonar, podemos empezar a describir las relaciones inferidas por el sistema, y las razones por las que el sistema incluiría los contenidos de la figura 3 en la recomendación final.

En primer lugar, tenemos una relación entre la retransmisión de Fórmula 1 vista por el usuario, y un documental deportivo en el que participan tanto el piloto “Fernando Alonso” (presente en el perfil del usuario), como el futbolista “Beckham”. En la figura 3 también se aprecia una relación entre una retransmisión de fútbol (instancia de la clase *Soccer Broadcasting*) y el estadio “Yokohama”, donde éste tiene lugar. Además en esta última cadena semántica directa también participa una instancia de la clase *Soccer Players*, por lo que el sistema descubre un nexo común entre el documental deportivo descubierto antes, y esta retransmisión de fútbol. Esta relación de tipo *cadena semántica directa con nodos comunes*, permite que el sistema infiera una asociación entre ambos programas, aún cuando no es el mismo futbolista en el que interviene en ambos. Este tipo de inferencias sólo tiene cabida en un entorno de razonamiento sobre la semántica de los contenidos de TV. En resumen, a partir de  $TC_{C_0}$ , el sistema infiere, entre otras, las siguientes relaciones:

---

*Formula 1 Broadcasting*  $\Rightarrow$  *Sport Documentaries*  
*Soccer Broadcasting*  $\Rightarrow$  *Sport Places*  
*Sport Documentaries*  $\Rightarrow$  *Soccer Players* *Soccer Broadcasting*  
*Formula 1 Broadcasting*  $\Rightarrow$  *Soccer Players* *Sport Soccer Broadcasting*

---

De los dos nuevos contenidos de TV descubiertos por el sistema (*soccer broadcasting* y *sport documentaries*), el segundo es especialmente relevante, ya que está relacionado con uno de los valores presentes en el perfil de usuario: el piloto de Fórmula 1. Sin embargo, el sistema también debe considerar en el proceso de recomendación, la importancia del programa sobre fútbol descubierto, debido a que este tipo de programas y las retransmisiones de Fórmula 1 (vistas por el usuario) son instancias de clases hermanas.

Sigamos descubriendo otras relaciones semánticas existentes en la figura 3. Podemos ver que hay una *cadena semántica directa* entre la película de acción vista por el usuario, y una película de terror, establecida a través del actor “Clint Eastwood”. Además esta película está relacionada con la instancia “Amsterdam Arena” de la clase *Sport Places*, ya que la trama de la película transcurre en este estadio. Esto permite descubrir un nexo de unión entre la mis-

ma y el partido de fútbol inferido anteriormente: ambas instancias están asociadas a lugares donde tienen lugar eventos deportivos (clase “Sport Places”), y por tanto hay una relación del tipo *cadena semánticas directas con nodos comunes* entre ambas.

La película de terror se incluirá en la recomendación final por los dos motivos que acabamos de presentar. Por una parte, interviene en ella el actor definido en el perfil de usuario, y por otro lado, está relacionado semánticamente con el partido de fútbol, cuya relevancia en el proceso de personalización ya ha sido explicada antes.

Las relaciones que hemos descrito se resumen así:

---

*Action Movies*  $\Rightarrow$  *Terror Movies*  
*Terror Movies*  $\Rightarrow$  *Sport Places*  
*Terror Movies*  $\Rightarrow$  *Sport Places Soccer Broadcasting*

---

Exploremos ahora las restantes clases del conjunto  $TC_{C_1}$ . Por ejemplo, podemos ver en la figura 3, una *cadena semántica directa* entre la telenovela y el concurso. Además dicha *cadena semántica directa*, permite descubrir sendas *cadena indirectas*. Veamos cómo: partimos de las siguientes *cadena semánticas directas*:

*Soap Operas* [ *hasActress* ] *Actresses* [ *presenterOf* ] *Quiz Shows*  
*Action Movies* [ *hasStarringActors* ] *Actors* [ *producerOf* ] *Terror Movies*

Con arreglo a la jerarquía de clases y propiedades establecida en la ontología, la propiedad *hasActress* está relacionada mediante *nexo semántico* con *hasStarringActors*, ya que la primera es hermana de la superpropiedad de la segunda, tal como se muestra en la figura 4.

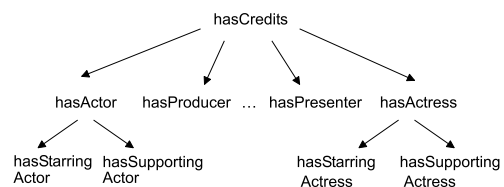


Figura 4: Subconjunto de la jerarquía de propiedades

Dado que existe un *nexo semántico* entre las propiedades *presenterOf* y *producerOf* (son propiedades hermanas), el sistema puede inferir una *relación indirecta* entre las instancias de las clases *Soap Operas* y *Action Movies*. La segunda relación indirecta mencionada antes se establecería de manera análoga entre las instancias de las clases *Soap Operas* y *Comedy Movies*. Es decir, se infieren:

---

*Soap Operas*  $\approx$  *Action Movies*  
*Soap Operas*  $\approx$  *Comedy Movies*

---

Considerando estas relaciones, la película de acción vista por el usuario estaría relacionada semánticamente con una telenovela. En principio, esta relación indirecta no involucra ningún valor concreto definido en el perfil personal del espectador, por lo que la telenovela no sería especialmente relevante de cara a la recomendación final. Sin embargo, si seguimos razonando sobre la ontología de la figura 3 vemos que es posible descubrir una asociación más directa con valores conocidos por el sistema.

Para ello, centrémonos en la instancia de la clase *Comedy Movies*. Este programa se puede relacionar semánticamente con la película de acción mencionada antes, ya que también en ella participa el actor favorito del usuario.

Además, podemos inferir una nueva cadena semántica entre la instancia de *Comedy Movies* y un concurso televisivo presentado por la actriz protagonista de la comedia. En la figura 3, es fácil ver que también se puede establecer una asociación semántica entre la telenovela y esta comedia, ya que ambas tienen un nexo en común: "Julia Roberts". Estas últimas relaciones descubiertas se resumen como:

---

---

*Comedy Movies*  $\Rightarrow$  *Actors Action Movies*  
*Comedy Movies*  $\Rightarrow$  *Quiz Shows*  
*Soap Operas*  $\Rightarrow$  *Quiz Shows*  
*Comedy Movies*  $\Rightarrow$  *Actresses Soap Operas*

---

---

A la vista de los contenidos sugeridos por AVATAR en este ejemplo, podemos concluir que este tipo de mecanismos de recomendación, basados en el establecimiento de asociaciones semánticas, son especialmente útiles para descubrir al usuario contenidos que no conocía. Estos programas, sin embargo, pueden ser de su interés, debido a que guardan relación, desde un punto de vista semántico, con las preferencias declaradas en su perfil personal.

## 6. Conclusiones y trabajo futuro

En este artículo se ha presentado el marco de razonamiento semántico de AVATAR, un sistema recomendador de contenidos de televisión en el que estamos trabajando, y cuyo principal objetivo es conseguir mejorar las recomendaciones personalizadas presentadas a los espectadores, utilizando para ello la experiencia de la Web Semántica. En dicho marco hemos incluido la implementación de una ontología OWL para representar el conocimiento del dominio de la TV, necesaria para soportar los procesos de inferencia. Además, hemos propuesto un mecanismo de representación de las preferencias de los usuarios, basado en la reutilización de la jerarquía de clases y propiedades, así como instancias específicas, definidas en dicha ontología. Así, nuestros *perfiles-ontología* pueden ser construidos incrementalmente a medida que el sistema recibe información adicional sobre las preferencias de los usuarios.

Sobre este marco se propone un algoritmo de razonamiento semántico que permite inferir relaciones entre programas a partir de la información almacenada en la ontología, con el objetivo de conseguir mejores recomendaciones que las proporcionadas por los recomendadores actuales basados en técnicas sintácticas.

Para validar el algoritmo de inferencia que proponemos hemos desarrollado un prototipo del sistema. Para completar este prototipo, debemos abordar como trabajo futuro el problema de ordenar las relaciones descubiertas en base a su relevancia semántica y a la información almacenada en el perfil de usuario. Esto se debe a que la capacidad de razonamiento del sistema AVATAR, permite inferir una gran cantidad de relaciones, pero no todas ellas son igual de importantes de cara a la recomendación. En este momento, nuestro trabajo se centra en este proceso de "ponderación". Para ello tendremos en cuenta factores como: (i) el tipo de relación

semántica: así por ejemplo, las *cadena semánticas directas* son relaciones más relevantes que las *indirectas*; (ii) la longitud de las cadenas semánticas, ya que las cadenas que involucran a muchas clases son relaciones menos directas y, por tanto, menos interesantes; (iii) la presencia de valores definidos en el perfil de usuario en las cadenas descubiertas; y (iv) la presencia de campos definidos en la especificación TV-Anytime que pueden ser relevantes a la hora de ofrecer una recomendación: por ejemplo, el campo *Intended Audience* que identifica el público al que va dirigido el programa.

## Referencias

- [1] Blanco Fernández Y., Pazos Arias J. J., Gil Solla A., Ramos Cabrer M., Barragáns Martínez B. and López Nores M. A Multi-Agent Open Architecture for a TV Recommender System: A Case Study using a Bayesian Strategy. In *Proc. of the IEEE Sixth International Symposium on Multimedia Software Engineering*, 2004.
- [2] Blanco Fernández Y., Pazos Arias J. J., Gil Solla A., Ramos Cabrer M., López Nores M. and Barragáns Martínez B. AVATAR: Modeling Users by Dynamic Ontologies in a TV Recommender System based on Semantic Reasoning. In *Proc. of the 3rd European Conference on Interactive Television: User Centred ITV Systems, Programmes and Applications (EuroITV-05)*, 2005.
- [3] Daconta M., Obrst L. J. and Smith K. J. *The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management*. John Wiley & Sons, 2003.
- [4] ETSI TS 101 812. *Multimedia Home Platform (MHP)*. 2002.
- [5] Geroimenko V. and Chen C. *Visualizing the Semantic Web*. Springer Verlag, 2003.
- [6] McGuinness D. and van Harmelen F. OWL Web Ontology Language Overview. W3C Recommendation. 2004.
- [7] Paganí M. *Multimedia and Interactive Digital TV: Managing the Opportunities Created by Digital Convergence*. Idea Group Publishing, 2003.
- [8] Ricci F., Arslan B., Mirzadeh N. and Venturini A. ITR: a Case-Based Travel Advisory System. In *Proc. of 6th European Conference on Case Based Reasoning (ECCBR-02)*, 2002.
- [9] Schonfeld E. Don't Just Sit There. Do Something. *Business 2.0*, 2000.
- [10] Staab S. and Studer R. *Handbook on Ontologies*. Springer, 2004.
- [11] Zimmerman J., Kurapati K., Buczak A. L., Schaffer D., Gutta S. and Martino J. TV Personalization System: Design of a TV Show Recommender Engine and Interface. *Personalized Digital Television: Targeting Programs to Individual Viewers*, 2004.