

**OEG Publication**

De Diego R, Fernández-López M, Gómez-Pérez A, Ramos JA

***OEGMerge: un modelo de mezcla de ontologías basado en casuísticas***

CAEPIA 2005: Conferencia de la Asociación Española para la Inteligencia Artificial  
Edited by Asociación Española para la Inteligencia Artificial.

November 2005.

Santiago de Compostela, Spain.

ISBN: 84-96474-13-5. Vol I. Páginas: 283-292

# OEGMerge: un modelo de mezcla de ontologías basado en casuísticas

Raúl de Diego, Mariano Fernández-López, Asunción Gómez-Pérez, José Ángel Ramos

OEG - Facultad de Informática. Universidad Politécnica de Madrid (UPM)  
Campus de Montegancedo, s/n. 28660 Boadilla del Monte. Madrid. Spain  
{mfernandez@fi.upm.es, asun@fi.upm.es, jarg@fi.upm.es}

**Resumen.** Desde hace tiempo la mezcla de ontologías es una actividad necesaria, sin embargo, los actuales métodos de mezcla de ontologías no tienen una casuística detallada ni una formalización precisa. Para la validación de estos métodos, es conveniente disponer de una casuística lo más completa posible. Por ello, en este artículo se presenta el modelo OEGMerge, desarrollado a partir de la experiencia del Grupo de Ingeniería Ontológica (OEG) de la UPM, en el que se describe detallada y formalmente la casuística de mezcla y las acciones a realizar en cada caso. En esta primera aproximación sólo se abarca taxonomía de conceptos, atributos y relaciones.

## 1. Introducción

Las ontologías permiten capturar conocimiento consensuado de un dominio dado de una forma genérica y formal, para ser reutilizado y compartido por aplicaciones y grupos de personas. Desde esta definición se puede inferir erróneamente que hay una sola ontología para modelar un dominio. Sin embargo, se puede encontrar en la literatura varias ontologías que modelan el mismo conocimiento de diferentes formas. Por ejemplo, hay varias ontologías *top-level* que difieren en el criterio seguido para clasificar los conceptos más generales de la taxonomía (ej. [7]), y en el campo del comercio electrónico hay varios estándares e iniciativas conjuntas para la clasificación de productos y servicios (UNSPSC<sup>1</sup>, e-cl@ss<sup>2</sup>, RosettaNet<sup>3</sup>, etc.). Esta heterogeneidad en ontologías también se presenta en otros muchos dominios (medicina, ley, arte, ciencias, etc.).

Para Noy y Musen [10] el alineamiento de ontologías (*ontology alignment*) consiste en el establecimiento de diferentes clases de *mappings* (o correspondencias) entre los componentes de dos ontologías, preservando con esto las ontologías originales; y la mezcla de ontologías (*ontology merging*) propone generar una única ontología a partir de las ontologías originales. Este artículo se va a centrar en la mezcla de ontologías, aunque algunas ideas presentadas puedan ser utilizadas para el

---

<sup>1</sup> <http://www.unspsc.org/>

<sup>2</sup> <http://www.eclass.de/>

<sup>3</sup> <http://www.rosettanel.org/>

alineamiento de las mismas. De hecho, antes de mezclar dos ontologías hay que establecer correspondencias entre los términos de las ontologías a mezclar.

Hasta ahora, aunque se han publicado algoritmos que permiten mezclar ontologías sencillas, los autores de estos algoritmos no han proporcionado las casuísticas que se pueden presentar a la hora de hacer mezcla, ni tan siquiera de taxonomías.

En el Grupo de Ingeniería Ontológica (OEG<sup>4</sup>) se ha elaborado, e implementado como un *plug-in* de WebODE [1], el modelo **OEGMerge**, para la mezcla de ontologías que especifica la casuística encontrada en las mezclas realizadas en diferentes proyectos. Una de las características más relevantes de este modelo es que no se trata realmente de un algoritmo, sino de un sistema basado en reglas. De hecho, las características del problema de mezcla (uso de conocimiento incompleto, de heurísticas, de conocimiento de ingenieros ontológicos, etc.) hacen pensar que un modelo basado en reglas es más adecuado que un algoritmo procedimental, teniendo como base de conocimiento las ontologías y el conjunto de *mappings* entre ellas.

En la sección 2 de este artículo se revisan los métodos de mezcla de ontologías propuestos, en la sección 3 se presenta el conocimiento previo del que se parte, formalizado de manera sencilla, en la sección 4 se presenta el modelo **OEGMerge** y en la sección 5 se presenta la implementación parcial de OEGMerge: ODEMerge. Por último, en la sección 6 se muestran las conclusiones y los trabajos futuros.

## 2. Estado de la cuestión sobre los métodos de mezcla de ontologías

Las contribuciones más importantes sobre cómo mezclar ontologías son ONIONS [6], FCA-Merge [14] y PROMPT [11].

*ONIONS*. Con este método se puede crear una biblioteca de ontologías provenientes de diferentes fuentes. En los primeros estadios de la integración, las fuentes son unificadas sin considerar las teorías generales que se usan en la modelización (teorías de parte-todo o desconectividad, por ejemplo). Después, estas teorías se incorporan como una base de la biblioteca y se integran como caras de un mismo poliedro. De esta manera, las ontologías de la biblioteca están conectadas a través de los términos de las teorías genéricas compartidas. Desde el punto de vista estricto no se puede considerar mezcla de ontologías puesto que no se genera un ontología única a partir de las fuentes, pero sí se trata la biblioteca completa como si fuera una ontología.

*FCA-Merge* [14]. Este enfoque es muy diferente al de los otros presentados en esta sección. FCA-Merge toma como entrada dos ontologías a mezclar y un conjunto de documentos del dominio de las ontologías. La mezcla se hace identificando los documentos que contienen información que se identifica como instancias que pertenecen a conceptos de ambas ontologías.

El método *PROMPT* [11]. La principal asunción de PROMPT es que las ontologías a mezclar están formalizadas con un modelo de conocimiento común basado en marcos. Existe un *plug-in* de Protégé-2000 mezcla ontologías de acuerdo al método PROMPT. Este método propone primero elaborar una lista de las operaciones a ser

---

<sup>4</sup> <http://www.oeg-upm.net>

realizadas para mezclar las dos ontologías (por ejemplo: mezclar dos clases, mezclar dos atributos, etc.). Esta actividad es llevada a cabo automáticamente por el *plug-in* de PROMPT. Acto seguido, comienza un proceso cíclico. En cada ciclo el ingeniero ontológico selecciona una operación de la lista y la ejecuta. Se genera entonces una lista de conflictos resultantes de la ejecución de la operación y la lista de operaciones posibles para las iteraciones siguientes se actualiza. Algunas de estas nuevas operaciones son incluidas en la lista porque son útiles para resolver los conflictos.

Las propuestas anteriormente mencionadas suponen un punto de inicio en la investigación sobre la mezcla de ontologías, sin embargo, sus autores no proporcionan detalles de las casuísticas que se presentan a la hora de llevar a cabo la mezcla. A lo largo de los apartados siguientes, se va a presentar OEGMerge, un modelo, implementado parcialmente en WebODE [1], que está detallado utilizando una serie de reglas basadas en los casos que se pueden presentar al mezclar taxonomías. Las premisas de estas reglas, serán los casos posibles a los que la mezcla se enfrenta.

Existen otros sistemas de mezcla de ontologías, pero que no representan métodos en sí, sino soluciones para un determinado lenguaje o para una determinada acción: OntoMerge<sup>5</sup>, CUPID [9], GLUE [4], etc.

### 3. Conocimiento previo

Para mostrar el modelo, primero se proporcionarán las definiciones de ontología y de conjunto de *mappings* en que se apoya. Así, esto funcionará como base de conocimiento del sistema basado en reglas.

La formalización de ontología que se expresa a continuación se basa en la definición de Studer y colegas [13]: una ontología es una especificación formal y explícita de una conceptualización compartida.

**Definición 1.-** Una ontología es una tupla  $O = (C, A, atributoDe_O, subclaseDe_O, R)$  donde  $C$  es un conjunto de conceptos,  $A$  es un conjunto de atributos,  $atributoDe_O \subseteq A \times C$ ,  $subclaseDe_O \subseteq C \times C$  y  $R \subseteq C \times C$  el conjunto de relaciones ad hoc.

Entre las observaciones a tener en cuenta sobre la definición 1, en primer lugar, hay que hacer notar que está pensada para describir la primera versión de OEGMerge. Para otros propósitos, sería necesaria una definición más amplia. En segundo lugar, las relaciones *atributoDe<sub>O</sub>* y *subclaseDe<sub>O</sub>* llevan la etiqueta O para indicar que está particularizadas a la ontología O. Por último, los términos “concepto”, “atributo” y “relación” se consideran primitivos a efectos de este artículo y, por tanto, no se van a definir formalmente aquí.

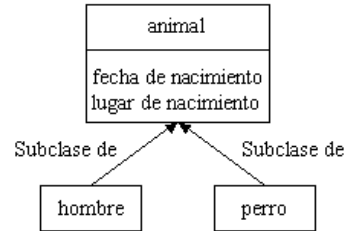
Por ejemplo, la ontología de la figura 1 se representa, de acuerdo con la definición 1, de la siguiente manera:

$$O_{animales} = (C = \{hombre, perro, animal\},$$

---

<sup>5</sup> <http://cs-www.cs.yale.edu/homes/dvm/daml/ontology-translation.html>

$A = \{\text{fecha de nacimiento},$   
 $\quad \text{lugar de nacimiento}\},$   
 $\text{atributoDe}O_{\text{animales}} =$   
 $\quad \{(\text{fecha de nacimiento}, \text{animal}),$   
 $\quad (\text{fecha de nacimiento}, \text{perro}),$   
 $\quad (\text{fecha de nacimiento}, \text{hombre}),$   
 $\quad (\text{lugar de nacimiento}, \text{animal}),$   
 $\quad (\text{lugar de nacimiento}, \text{perro}),$   
 $\quad (\text{lugar de nacimiento}, \text{hombre})\},$   
 $\text{subclaseDe}O_{\text{animales}} =$   
 $\quad \{(\text{hombre}, \text{animal}), (\text{perro}, \text{animal})\},$   
 $R = \emptyset$



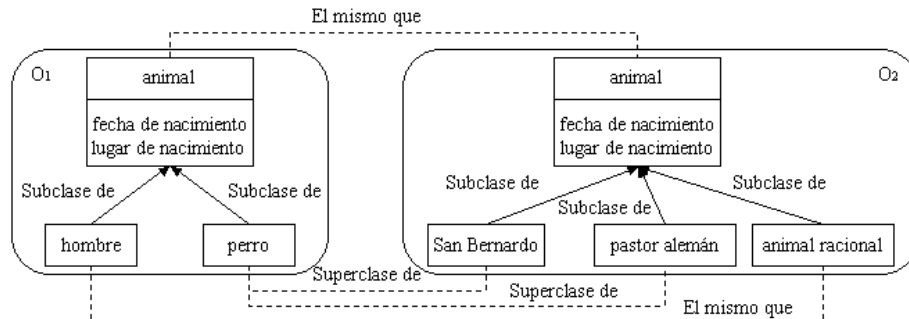
**Figura 1: Ejemplo de ontología de animales**

**Definición 2.-** El conjunto de *mappings* entre las ontologías  $O_1$  y  $O_2$  es una tupla  $M = (O_1, O_2, \text{elMismoQue}_M, \text{subclaseDe}_M, \text{superclaseDe}_M)$  donde  $\text{elMismoQue}_M \subseteq \mathcal{P}(T_1) \times \mathcal{P}(T_2)$  y  $\text{subclaseDe}_M, \text{superclaseDe}_M \subseteq C_1 \times C_2$ , siendo  $C_1$  el conjunto de conceptos de  $O_1$ , y  $C_2$  el conjunto de conceptos de  $O_2$ , y siendo  $T_1 = C_1 \cup A_1 \cup R_1$  y  $T_2 = C_2 \cup A_2 \cup R_2$  los términos de  $O_1$  y  $O_2$ .

Obsérvese que la relación  $\text{elMismoQue}_M$  se establece entre cualquier término, por ejemplo, entre un atributo y una relación, o incluso entre conjuntos de términos (por eso se hace referencia a  $\mathcal{P}$ , todos los subconjuntos posibles en el conjunto de términos). Así, por ejemplo, un atributo de una ontología, como es el caso de *nombre*, puede corresponderse con varios atributos de la otra ontología: *nombre*, *apellido 1* y *apellido 2*.

A modo de ejemplo, el conjunto de los *mappings* de la figura 2 se representa, de acuerdo con la definición 2, de la siguiente manera:

$M_{\text{animales}} = (O_1, O_2, \text{elMismoQue}M_{\text{animales}} = \{(\text{animal}, \text{animal}),$   
 $\quad (\text{hombre}, \text{animal racional})\},$   
 $\text{subclaseDe}M_{\text{animales}} = \emptyset,$   
 $\text{superclaseDe}M_{\text{animales}} = \{(\text{perro}, \text{San Bernardo}),$   
 $\quad (\text{perro}, \text{pastor alemán})\}$



**Figura 2: Ejemplo de mappings entre dos ontologías**

Al igual que ocurría con la definición 1, aquí también se está considerando sólo lo imprescindible para definir nuestro modelo OEGMerge. Además, según la definición 2, un *mapping* establece relaciones de identidad, de *superclase de* y de *subclase de* entre los conceptos de la ontología  $O_1$  y de la ontología  $O_2$ . En futuros trabajos, se extenderá la semántica para abarcar etiquetas como *unión*, *parte de*, etc. Por último, la generación automática de estos *mappings*, utilizando similitudes entre etiquetas, documentos, estructuras, instancias, etc. está fuera del alcance de este artículo (véanse, por ejemplo, [2] o [5]).

Una vez que se ha definido qué es una ontología y el conjunto de *mappings* entre las ontologías, se está en condiciones de definir el modelo de mezcla OEGMerge.

#### 4. OEGMerge

El modelo OEGMerge surge de la experiencia del Grupo. Esta experiencia realizada metódicamente ha dado lugar a la identificación de una casuística, detallada en este artículo. El modelo parte de dos ontologías originarias para producir una nueva ontología que contenga todo el conocimiento de las dos originarias.

Es **importante** tener en cuenta que el modelo aquí expuesto no considera ontologías con contradicciones, por ejemplo, que en una se modelice que  $a$  es subclase de  $b$ , y en otra que  $b$  es subclase de  $a$ . Por esta razón, se recomienda llevar a cabo una tarea de evaluación antes de realizar la mezcla. También hay que considerar el hecho de que las casuísticas que se exponen son fruto de la experimentación y, por consiguiente, no son necesariamente exhaustivas.

Las dos ontologías originales se van a identificar a partir de ahora como ontología base y ontología auxiliar. La elección de la ontología base y, por tanto, de la auxiliar no influye en el resultado final de la mezcla. Se realiza la diferencia sólo para tomar una de las ontologías como punto de referencia. No obstante, la experiencia recomienda elegir como ontología base la más completa por número de términos y atributos.

**Definición 3.-** La mezcla de dos ontologías  $O_{aux}$ ,  $O_{base}$  utilizando un conjunto de *mappings*  $M$  es una función *MezclaOntología*:  $O_{aux} \times O_{base} \times M \rightarrow O_{res}$  que obtiene la ontología resultado después de aplicar los siguientes pasos:

- Paso 1) Se establece  $O_{res} = O_{base}$   
Paso 2) Se ejecutan las reglas que se muestran en la Tabla 1, que, a su vez, ejecuta la acción *MezclaIdentidad*. Esta acción también ejecuta una serie de reglas: las mostradas en la Tabla 2.

En todas las tablas se asume que la **estrategia de control de ejecución de las reglas es el orden de numeración** dentro de la tabla.

Las condiciones de la casuística de las tablas 1 y 2 se muestran en la Tabla 3, y las acciones, en la Tabla 4.

**Tabla 1: Reglas de definición de la función *MezclaOntología***

Caso	Condición	Acción
1	$elMismoQue_M(c_{aux}, c_{base})$	$MezclaIdentidad(c_{aux}, c_{base}, O_{aux}, O_{base}, mapping)$
2	$superclaseDe_M(c_{aux}, c_{base})$	$insertarSubclaseDe(c_{base}, c_{aux})$
3	$subclaseDe_M(c_{aux}, c_{base})$	$insertarSubclaseDe(c_{aux}, c_{base})$

La regla 1 establece que si el concepto de la ontología auxiliar es el mismo, según el *mapping*, que el de la ontología base, entonces se lleva a cabo la acción *MezclaIdentidad*, definida en la Tabla 2. Los predicados que aparecen en las condiciones están definidos en la Tabla 3, y los de las acciones están definidos en la Tabla 4. Las reglas 2 y 3 trasladan a la ontología resultado la relación *subclase de* establecida en el *mapping*. Aquí es importante tener en cuenta que, en caso de estar trabajando con una herramienta gráfica, hay que borrar las relaciones taxonómicas redundantes que surjan después de introducir nuevos enlaces *subclase de*. Así, por ejemplo, si se tiene  $subclaseDe(a, b)$ ,  $subclaseDe(b, c)$  y  $subclaseDe(a, c)$ , en la herramienta gráfica habría que borrar  $subclaseDe(a, c)$ .

**Tabla 2: Reglas de definición de la función *MezclaIdentidad***

Caso	Condición	Acción
I-1	$atributoNuevoAuxiliar(a_{aux}, c_{res}, O_{res}, mapping)_i$	$insertarAtributo(a_{aux}, c_{res}, O_{res})$
I-2	$atributosConcretosAuxiliar(atributos_{aux}, a_{base}, c_{res}, O_{res})$	$insertarConjuntoAtributos(atributos, c_{res}, O_{res})$ $borrarAtributo(a_{base}, c_{res}, O_{res})$
I-3	$atributosConcretosBase(atributos_{base}, a_{aux})$	<< Sin acción >>
I-4	$relacionNuevaAuxiliar(r_{aux}, c_{res}, O_{res}, mapping)_i$	$insertarRelación(r_{aux}, c_{res}, O_{res}, O_{aux})$
I-5	$atributoBaseEsRelacionAuxiliar(a_{aux}, r_{base})$	$insertarRelación(r_{aux}, c_{res}, O_{res}, O_{aux})$ $borrarAtributo(a_{base}, c_{res}, O_{res})$
I-6	$atributoAuxiliarEsRelaciónBase(a_{base}, r_{aux})$	<< Sin acción >>
I-7	$ascendentesCorrespondientes(c_{base}, c_{base}', c_{aux}, c_{aux}')$	<< Sin acción >>
I-8	$ascendenteHermano(c_{base}', c_{aux}, c_{aux}', O_{aux})$	$insertarSubclaseDe(c_{aux}, c_{base}')$

I-9	ascendentesNoCorrespondientes ( $c_{base}$ , $c_{base}'$ , $c_{aux}$ , $c_{aux}'$ )	insertarSubclaseDe ( $c_{base}$ , $c_{base}'$ )
-----	---	---

Los tres primeros casos de la Tabla 2 se corresponden con las situaciones ante los atributos que se pueden encontrar en la ontología auxiliar: existencia de un atributo que no existe en la base (caso I-1), existencia de atributos auxiliares que concretan uno base (caso I-2) y existencia de atributos base que concretan uno auxiliar (caso I-3).

Los siguientes tres casos corresponden a las situaciones que se dan en la mezcla de relaciones: existencia de una relación auxiliar que no existe en la base (caso I-4), existencia de una relación auxiliar que es atributo base (caso I-5) y existencia de atributo auxiliar que es relación base (caso I-6).

Los tres últimos casos que recoge la tabla son los correspondientes a las diferencias en las relaciones taxonómicas, siendo éstos: existencia de ascendente común (concepto base y auxiliar que tengan correspondencia) (caso I-7), existencia de un ascendente auxiliar que es hermano en la ontología base (caso I-8) y existencia de un ascendente auxiliar que no corresponde con ningún ascendente base (caso I-9). Este último caso se corresponde con la aparición de herencia múltiple como producto de la mezcla, no existente en las ontologías originales.

Es importante tener en cuenta que se asume que es preferible considerar varios atributos en vez de uno sólo que los sintetice a todos (casos I-2 e I-3). Además, se ha considerado que es preferible establecer relaciones entre dos conceptos, que un atributo cuyo tipo es un concepto (casos I-5 e I-6). Estas acciones se han determinado así por la experiencia del Grupo en las mezclas realizadas. Puesto que quizás no todos los investigadores estén de acuerdo con todas estas suposiciones, pueden hacer los cambios oportunos en el procedimiento de mezcla, modificando la acción a realizar en cada caso. Aun así, los casos que se consideran aquí serían los mismo, sólo implicando cambio en su tratamiento.

En la columna de acción de la Tabla 2 aparece en cuatro ocasiones “<<Sin Acción>>”. En la mezcla, según los criterios seguidos, no ha de realizarse ninguna acción para estos casos. Sin embargo, puesto que lo que se pretende es detallar toda la casuística que la mezcla va a encontrar, hemos reflejado el caso aunque no conllevara la realización de una acción.

**Tabla 3: Predicados de las condiciones**

Predicado	Definición formal
atributoNuevoAuxiliar( $a$ , $c$ , $o$ , mapping)	$\neg$ atributoDe $_O(a, c) \wedge$ $\neg \exists$ términos $tq$ elMismoQue $_M(\{a\}, \text{términos})$
atributosConcretosAuxiliar( $atributos_{aux}$ , $a_{base}$ )	elMismoQue $_M(atributos_{aux}, a_{base}) \wedge$ $\exists c \in o_{aux} tq$ atributoDe $_{O_{base}}(a, c)$ $\forall a \in$ atributos
atributosConcretosBase( $atributos_{base}$ , $a_{aux}$ )	elMismoQue $_M(atributos_{base}, a_{aux}) \wedge$ $\exists c \in o_{base} tq$ atributoDe $_{O_{aux}}(a, c)$ $\forall a \in$ atributos
relaciónNuevaAuxiliar( $r$ , $c$ , $o$ , mapping)	$\neg \exists c'$ relaciónEntre $_O(r, c, c') \wedge$ $\neg \exists$ términos $tq$ elMismoQue $_M(\{r\}, \text{términos})$
atributoAuxiliarEsRelaciónBase( $a_{base}$ , $r_{aux}$ )	esMismoQue $_M(a_{base}, r_{aux})$



atributoBaseEsRelacionAuxiliar( $a_{aux}, r_{base}$ )	esMismoQue $_M(a_{aux}, r_{base})$
ascendentesCorrespondientes( $c_{base}, c_{base'}$ , $c_{aux}, c_{aux'}$ )	elMismoQue $_M(c_{base'}, c_{aux'}) \wedge$ subclaseDe $_{O_{aux}}(c_{aux}, c_{aux'}) \wedge$ subclaseDe $_{O_{base}}(c_{base}, c_{base'})$
DifusiónDeNiveles( $c_{base}, c_{base'}$ , $c_{aux}, c_{aux'}$ )	$\neg$ subclaseDe $_{O_{base}}(c_{base}, c_{base'}) \wedge$ elMismoQue $_M(c_{base'}, c_{aux'}) \wedge$ subclaseDe $_{O_{aux}}(c_{aux}, c_{aux'})$
ascendenteHermano( $c_{base'}$ , $c_{aux}, c_{aux'}$ , $O_{aux}$ )	$\neg$ subclaseDe $_{O_{base}}(c_{base}, c_{base'}) \wedge$ elMismoQue $_M(c_{base'}, c_{aux'}) \wedge$ subclaseDe $_{O_{aux}}(c_{aux}, c_{aux'}) \wedge$ subclaseDe $_{O_{aux}}(c_{aux'}, c_{aux'})$

La Tabla 3 recoge la formalización de los predicados que aparecen en las condiciones de las tablas de reglas (tablas 1 y 2). En la Tabla 4 se recoge la formalización de los predicados que aparecen en las acciones de las tablas de reglas (tablas 1 y 2).

**Tabla 4: Predicados de las acciones**

Acción	Definición informal	Definición formal
insertarSubclaseDe( $c, c'$ )	Se inserta un enlace de <i>subclase de</i> entre dos conceptos.	subclaseDe $_{O_{i+1}} = \text{subclaseDe}_{O_i} \cup \{(c, c')\}$ $C_{O_{i+1}} = C_{O_i} \cup \{c, c'\}$
insertarAtributo( $a, c, o$ )	Se añade el atributo $a$ al concepto $c$ de la ontología $o$ .	atributoDe $_{O_{i+1}} = \text{atributoDe}_{O_i} \cup \{(a, c)\}$ ; $A_{O_{i+1}} = A_{O_i} \cup a$
borrarAtributo( $a, c, o$ )	Se borra el atributo $a$ del concepto $c$ de la ontología $o$ .	atributoDe $_{O_{i+1}} = \text{atributoDe}_{O_i} - \{(a, c)\}$ ; $A_{O_{i+1}} = A_{O_i} - a$
insertarConjuntoAtributos (atributos, $c, o$ )	Se añade cada uno de un conjunto de atributos al concepto $c$ de la ontología $o$ .	atributoDe $_{O_{i+1}} = \text{atributoDe}_{O_i} \cup \{(a, c) \mid a \in \text{atributos}\}$ ; $A_{O_{i+1}} = A_{O_i} \cup \{(a, c) \mid a \in \text{atributos}\}$
insertarRelación( $r_{aux}, c_{res}, O_{res}, O_{aux}$ )	Se añade la relación $r_{aux}$ al concepto $c_{res}$ de la ontología $O_{res}$ considerando el destino que tiene la relación en la ontología $O_{aux}$ .	relaciónEntre $_{O_{i+1}} = \text{relaciónEntre}_{O_i} \cup \{(r_{aux}, c_{res}, c') \mid (r_{aux}, c_{res}, c') \in \text{relaciónEntre}_{O_{aux}}\}$ ; $R_{O_{i+1}} = R_{O_i} \cup r_{aux}$

El modelo OEGMerge, tanto de manera manual como automática (véase el apartado 5) se ha utilizado desde el año 2001 [3; 12] para mezclar ontologías en el Grupo de Ingeniería Ontológica. La implementación parcial del método, ODEMerge, está disponible para el uso de todos los usuarios de WebODE, más de 200 repartidos por todo el mundo. Incluso los resultados de esta implementación se usan para trabajos externos, como [8].

## 5. ODEMerge

Con la realización de numerosas mezclas de ontologías siguiendo el modelo OEGMerge y utilizando la técnica descrita con el ejemplo del apartado 4, apareció la necesidad de eliminar en la medida de lo posible el proceso de edición correspondiente a la modificación de la ontología base con el contenido de la ontología auxiliar. Para ello, se desarrolló un servicio integrado en la plataforma WebODE: ODEMerge.

ODEMerge [12] es un servicio que realiza la mezcla automática de las taxonomías de dos ontologías a partir de las ontologías y de dos tablas, que generará el ingeniero ontológico, con información semántica sobre sinonimia e hiperonimia entre los conceptos de las ontologías (información semántica que tiene como entrada el servicio equivalen a la información aportada por el experto del dominio en el método, es decir, el conjunto de *mappings*). Los detalles referentes a formato de las tablas y limitaciones del servicio se pueden encontrar en la ayuda de la plataforma y en [12].

Así, ODEMerge pretende realizar, en la medida de lo posible y de una manera automática, las acciones de la mezcla taxonómica del modelo OEGMerge. La automatización conlleva la eliminación de la copia de una ontología para tomarla como base, puesto que ya no se trabaja sobre una de las ontologías para producir la unificada, sino que la ontología unificada se genera independientemente. Además, en la última versión de este servicio se ha incorporado a la automatización la inserción de atributos de clase sobre el componente de la ontología original del que proviene el concepto, lo que proporciona trazabilidad al proceso mezcla.

## 6. Conclusiones y trabajos futuros

Con el trabajo aquí presentado mediante el modelo OEGMerge, el Grupo de Ingeniería Ontológica establece de forma explícita la casuística fruto de su experiencia en la mezcla de ontologías, y establece las acciones resolutorias para llevar a cabo el proceso para el caso reducido de conceptos, atributos y relaciones (tres de los componentes fundamentales de toda ontología bajo el paradigma de marcos).

Además, se ha presentado el software desarrollado para la automatización de parte del modelo sobre una plataforma de ingeniería ontológica como es WebODE, cubriendo así otra tarea más del ciclo de vida de las ontologías. Tanto el modelo como el software presentado se han probado desde su desarrollo en numerosos procesos de mezcla llevados a cabo por el Grupo de Ingeniería Ontológica de la UPM.

Queda para un trabajo futuro el seguir ampliando tanto OEGMerge como ODEMerge para conseguir abarcar el resto de componentes de ontologías que ahora no abarca (axiomas, referencias, grupos, etc.) tanto bajo el paradigma de marcos como otros. Además, como se ha indicado ya, habría que contemplar en la definición de los *mappings* predicados nuevos para expresar correspondencias con más semántica.

## 7. Agradecimientos

La formalización del modelo ha sido financiada por el proyecto Infraestructura tecnológica de servicios semánticos para la web semántica (TIN-2004-02660), del Ministerio de Ciencia y Tecnología. También agradecemos los comentarios realizados a María del Carmen Suárez de Figueroa Baonza y Miguel Esteban Gutiérrez.

## 8. References

1. Arpírez JC, Corcho O, Fernández-López M, Gómez-Pérez A (2003) *WebODE in a nutshell*. AI Magazine, 24(3)-37-47
2. Cañadas G, Fernández-López M, García-García R, Lama M, Sánchez-Alberca A, Sorzano COS (2004) *Framework for automatic generation of ontology mappings*. In: Dieste O (ed) Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (IISIC'04). Madrid, Spain
3. De Diego R (2001) *Método de mezcla de catálogos electrónicos*. Proyecto final de carrera, Facultad de Informática, Universidad Politécnica de Madrid. 2001.
4. Doan A, Madhavan J, Domingos P, Halvey A (2002) *Learning to map between ontologies on the semantic web*. In Proceedings of 11<sup>th</sup> International WWW Conference, Hawaii (2002)
5. Ehrig M, Sure Y (2004) *Ontology mapping - an integrated approach*. In Christoph Bussler, John Davis, Dieter Fensel, and Rudi Studer, editors, Proceedings of the First European Semantic Web Symposium, volume 3053 of Lecture Notes in Computer Science, pages 76--91, Heraklion, Greece, 2004. Springer Verlag.
6. Gangemi A, Pisanelli DM, Steve G (1999) An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. *Data & Knowledge Engineering* 31(2):183-220
7. Guarino N, Welty C (2000) A Formal Ontology of Properties. In: Dieng R, Corby O (eds) 12<sup>th</sup> International Conference in Knowledge Engineering and Knowledge Management (EKAW'00). Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 97-112
8. Gutiérrez L (2004) *Ontology-based semantic querying of the Web with respect to food recipes*. IMM-Thesis. Technical University of Denmark. ISSN: 1601-233X
9. Madhavan, J., Bern-stein, P. A., Rahm, E.: Generic schema matching with Cupid. *VLDB Journal* (2001) 49-58
10. Noy NF, Musen MA (1999) *SMART: Automated Support for Ontology Merging and Alignment* In: Gaines BR, Kremer B, Musen MA (eds) 12<sup>th</sup> Banff Workshop on Knowledge Acquisition, Modeling, and Management, Banff, Alberta, Canada, 4-7:1-20
11. Noy NF, Musen MA (2000) *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment* Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000). Austin, Texas. 2000.
12. Ramos JA (2001) *Mezcla automática de catálogos electrónicos*. Proyecto final de carrera, Facultad de Informática, Universidad Politécnica de Madrid. 2001.
13. Studer R, Benjamins VR, Fensel D (1998) *Knowledge Engineering: Principles and Methods*. IEEE Transactions on Data and Knowledge Engineering 25(1-2):161-197.
14. Stumme G, Maedche A (2001) *FCA-Merge: Bottom-Up Merging of Ontologies* IJCAI 2001 - Proceedings of the 17<sup>th</sup> International Joint Conference on Artificial Intelligent. Seattle, USA, Agosto 2001.