

# Database Reverse Engineering: Proposal of an Open Tool Based on a Semantic Model

M. Colman, G. Larriera, R. Ruggia

siLAB Laboratorio de Sistemas de Información  
Facultad de Informática del Instituto Universitario de Ciencias de la Información  
e-mail: silab @ ei.edu.uy - Web: <http://www.ei.edu.uy/silab/>

**Abstract.** *Database reverse engineering (DBRE) is made of techniques that permits the construction of a conceptual definition of a database, e.g. an Entity Relationship Model (ERM) from a relational database. DBRE techniques can be applied to solve many problems, for instance rebuilding and/or updating lost or unexistent database documentation, being used as a pivot in a data migration process, or as an aid in data mining and data extraction in poorly-documented databases. Due to its many applications, DBRE has gained attention from the database scientific community, originating several proposals.*

*Existing works about DBRE can be classified according to different parameters, for instance: by the power of proposed algorithms, whether the methods try to reengineer specific cases or general application cases, and according to the assumptions concerning availability of information about the database. Despite those differences, the proposals have in common: the use of ERM when representing process output, orientation to solve database redocumentation problems, and the specification of algorithms rather than the specification of tools.*

*This paper presents a DBRE tool which builds a semantic specification from a relational database structure, its SQL operations and its data properties. The most relevant characteristics of this tool are: first of all, it generates a semantic model specification which leads to derive other models, including ERM and Dimensional Models. Second, this tool is useful not only to redocument databases, but also to be the basis of database mining tools. Finally, the tool integrates output of algorithms based in differents techniques.*

**Keywords:** *Database Reverse Engineering, Databases, Reverse Engineering, Semantic Models, Semantic Discovery, Relational Model, Conceptual Design, Entity Relationship Model.*

# Ingeniería Reversa de Bases de Datos: Propuesta de Herramienta Abierta Basada en Modelo Semántico

M. Colman, G. Larriera, R. Ruggia

siLAB Laboratorio de Sistemas de Información  
Facultad de Informática del Instituto Universitario de Ciencias de la Información  
e-mail: silab @ ei.edu.uy- Web: <http://www.ei.edu.uy/silab/>

**Resumen:** *La Ingeniería Reversa de Bases de Datos es el conjunto de técnicas que permite la obtención de una representación conceptual de un esquema de base de datos a partir de su codificación. Sus aplicaciones son múltiples, desde la re-documentación de bases de datos que evolucionaron en el ambiente operativo hasta la reutilización de esquemas de bases de datos, pasando por el apoyo a la migración y la construcción de metabases.*

*Este artículo presenta una herramienta de DBRE que permite construir una especificación conceptual de una base de datos relacional a partir de su estructura, código de operaciones SQL y propiedades de sus datos. Las características más relevantes de esta herramienta son las siguientes. Primeramente, captura la semántica de la base de datos usando un Modelo Semántico independiente del uso que se dará a la especificación semántica, lo cual permite derivar otras en una variedad amplia de modelos, por ejemplo Modelo Entidad-Relación y Modelos Multidimensionales. Segundo, está orientada, no solo a re-documentar bases de datos, sino también a servir como base para herramientas de exploración de bases de datos. Finalmente, integra los resultados de algoritmos basados en diferentes técnicas.*

**Palabras Clave:** *Database Reverse Engineering, Databases, Reverse Engineering, Semantic Models, Semantic Discovery, Relational Model, Conceptual Design, Entity Relationship Model.*

## 1 Introducción

La Ingeniería Reversa de Bases de Datos (DBRE) consiste en un conjunto de técnicas que permiten construir una descripción conceptual correspondiente a una base de datos, por ejemplo un modelo Entidad-Relación (ERM) a partir de una base Relacional.

La DBRE puede ser aplicada para resolver diferentes problemas, por ejemplo: reconstruir y/o actualizar documentación perdida o inexistente de bases de datos, servir como pivot en un proceso de migración de datos, y ayudar en la exploración y extracción de datos en bases poco documentadas. Debido a sus múltiples aplicaciones, la DBRE ha concitado la atención de la comunidad científica en el área de Bases de Datos, dando origen una cantidad importante de propuestas.

Los trabajos existentes en DBRE pueden clasificarse según diferentes parámetros, por ejemplo: la potencia de los algoritmos propuestos, si se trata de métodos para *re-ingenierizar* casos específicos o cuadros de aplicación general, y según las hipótesis de las cuales partan en cuanto a la información disponible sobre la base de datos. A pesar de estas diferencias, las propuestas realizadas tienen en común: el uso del ERM para representar el resultado del proceso, la orientación a resolver problemas de re-documentación de una base de datos, y la especificación de algoritmos puntuales más que la especificación des herramientas.

En este artículo se presentan algunos resultados obtenidos en un proyecto en curso en el área de Reingeniería de Bases de Datos. Este proyecto apunta a construir una herramienta para obtener una especificación conceptual/semántica a partir de un esquema de base de datos operativo. A diferencia de

otras propuestas, la especificación conceptual/semántica no se limita a estructuras Entidad Relación, sino a generar esquemas de otros tipos de modelos o servir como guía para consultar la base de datos. La especificación conceptual/semántica construida es el centro de la herramienta y funciona como una base de conocimiento a partir de la cual se pueden derivar diferentes representaciones conceptuales, entre ellas, esquemas Entidad-Relación.

Las técnicas aplicadas para construir la especificación conceptual/semántica se basan en el análisis del esquema, de las operaciones SQL habitualmente aplicadas, así como de los datos de la base. Las técnicas usadas integran y extienden otras ya existentes. Principalmente se integran técnicas que identifican atributos clave de relaciones y que identifican dependencias funcionales y de inclusión. Las extensiones corresponden a la detección de relaciones de agrupamiento de objetos que pueden ser usadas para derivar jerarquías de dimensiones en un modelo multidimensional. Además se mantienen referencias a los datos fuente, lo cual permite usar la especificación conceptual/semántica como base para realizar *browsing* así como para la construcción de una metabase de la base de datos fuente.

Con respecto a la herramienta propuesta, se trata de incluir diferentes tipos de algoritmos basados en diferentes hipótesis (por ejemplo, que le dan más o menos peso a la información estructural), de forma de poder aplicar el más adecuado según las características de la base a re-ingenierizar. Esto permite la extensibilidad de la herramienta con respecto a nuevos algoritmos de re-ingeniería. La herramienta se está implementando en Prolog (los algoritmos de re-ingeniería) y en Visual Basic (las interfaces usuario).

Las principales contribuciones de este trabajo residen en: (i) la construcción de una herramienta que integra y extiende algoritmos propuestos en la bibliografía, (ii) la propuesta de un modelo conceptual/semántico que permite la derivación de diferentes tipos de objetos y que se usa para representar el resultado de la re-ingeniería.

El trabajo presentado ha sido desarrollado en el contexto del proyecto “Técnicas y Herramientas para Ingeniería Reversa de Bases de Datos” del Laboratorio de Sistemas de Información - Facultad de Informática.

El resto del artículo se estructura de la forma siguiente. La sección 2 presenta una visión general de trabajos en el área. La sección 3 presenta la herramienta propuesta, junto con el modelo conceptual/semántico. La sección 4 presenta conclusiones y trabajo futuro.

## 2 Panorama del estado del arte

La evolución de la ingeniería reversa de bases de datos puede dividirse en dos generaciones [CA97], que se diferencian claramente según los modelos de datos usados como entrada a los procesos de reingeniería, las metodologías utilizadas y las precondiciones requeridas para realizar la reingeniería.

En la primera generación (hasta comienzos de los años 90s), las propuestas se basaron principalmente en aplicar transformaciones a esquemas físicos de bases de datos –básicamente modelos navegacionales y sistemas convencionales de archivos- y a esquemas lógicos relacionales. En esta primera generación, las diversas propuestas tienden a automatizar al máximo el proceso de reingeniería, asumiendo restricciones fuertes –y muchas veces difíciles de lograr en situaciones reales- sobre la entrada.

En la segunda generación (desde comienzos de los 90s y hasta el presente) aparece una mayor preocupación en disponer de metodologías que permitan obtener modelos conceptuales de gran riqueza de expresión, partiendo de bases de datos relacionales con precondiciones mínimas. Por ejemplo, se asume que el esquema físico cumple con formas normales inferiores a 3NF y que los nombres de las columnas de las tablas, en principio, no aportan semántica alguna fiable [PTB\*96]. Las diversas propuestas de esta segunda generación, en su mayoría, utilizan como entrada: al esquema físico de la base, a las aplicaciones que acceden a los datos, a la extensión de la base de datos (i.e. los datos existentes) y a la experiencia del usuario, a los efectos de construir un modelo conceptual. El modelo conceptual resultante más frecuentemente abordado es el modelo extendido de entidades-relaciones (EER), una extensión al modelo

clásico de Entidad-Relacione (ER) [Che76]. Entres tales propuestas se mencionan [CA97] [PTB\*96] [Chi95] [HTJ\*94]. Otras propuestas derivan un modelo conceptual orientado a objetos o EERs con extensiones basadas en objetos [RH97] [And94].

Las propuestas más recientes utilizan información sobre el esquema físico (tablas, vistas e índices) disponible en el catálogo de la base de datos, asumiendo que las bases se encuentran poco normalizadas. Esta presunción se sustenta en la práctica general de los implementadores de desnormalizar las bases de datos a los efectos de ganar eficiencia en las operaciones de join. Adicionalmente, en muchos casos reales es frecuente que la presencia de nombres de atributos iguales no necesariamente representan el mismo concepto. Por lo tanto, también es frecuente que los procesos de re-ingeniería deban estar atentos a tal situación y no asumir nada de antemano respecto a los nombres de atributos utilizados en el esquema físico.

El análisis de las operaciones de acceso a los datos resulta de vital importancia en el proceso de detección de vínculos entre las entidades existentes. Por ejemplo, la propuesta de [PTB\*96] se basa en las consultas de tipo equi-join que aparezcan en el SQL de las aplicaciones y en conteo de datos a los efectos de obtener más información sobre los vínculos detectados entre entidades. La detección de propiedades de los datos, como ser dependencias funcionales y dependencias de inclusión, se realiza mediante la exploración de los datos disponibles en la extensión de la base de datos. Finalmente, es importante considerar que la presencia del usuario parece ser necesaria a los efectos de confirmar o rechazar resultados obtenidos por los algoritmos durante el proceso de reingeniería.

## 3 La herramienta de DBRE

### 3.1 Arquitectura

La herramienta de DBRE consta de cinco partes principales (ver Fig. 1):

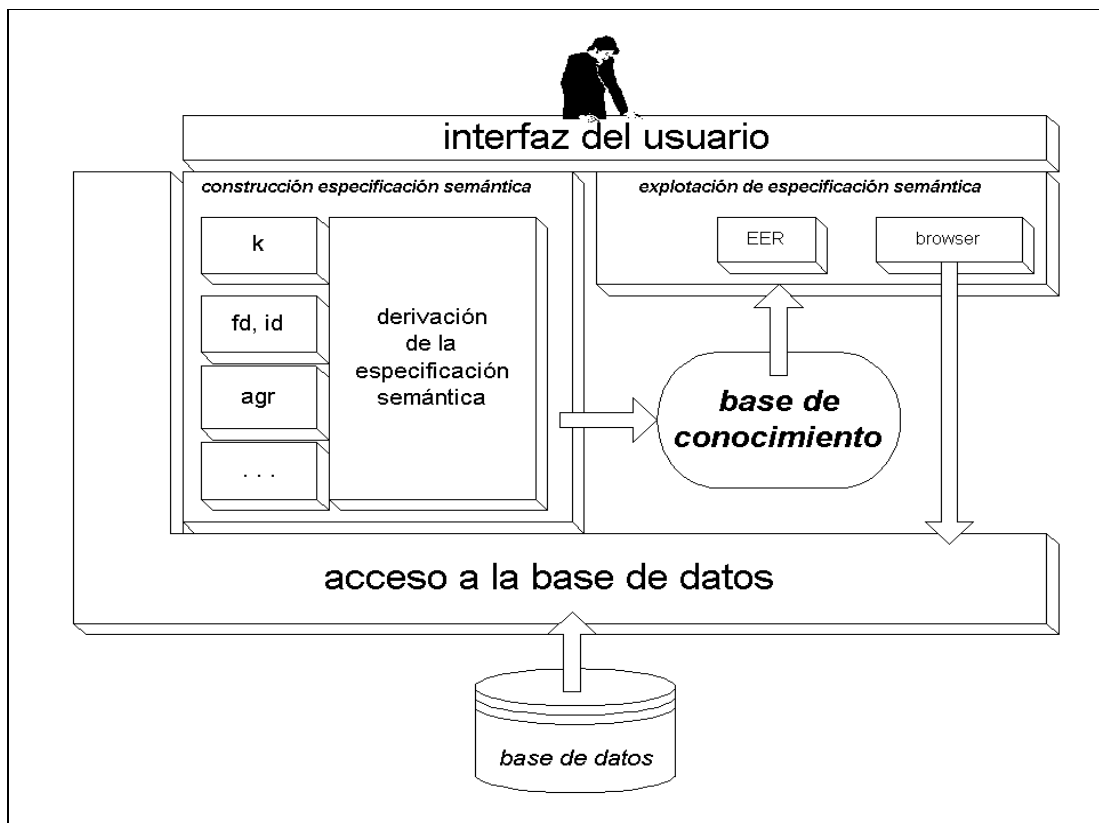
**Los módulos de construcción de la especificación semántica.** Estos módulos analizan la base de datos fuente y construyen la especificación semántica. Se distinguen dos tipos de módulos: (i) los algoritmos de re-ingeniería, que detectan información específica (claves, dependencias, agrupamientos), y (ii) el algoritmo de derivación de la especificación semántica, que toma como entrada la información generada por los primeros y construye la especificación correspondiente.

**La base de conocimiento.** Esta base de datos almacena la información generada durante el proceso de re-ingeniería. Dicha información se clasifica en dos tipos: (i) la especificación semántica de la base de datos analizada, y (ii) datos intermedios (llamados *asepciones de trabajo*) generadas por los algoritmos de re-ingeniería. Estos últimos son los que utiliza el algoritmo de derivación de la especificación semántica para construirla.

**Los módulos de explotación de la especificación semántica.** Estos módulos permiten explotar la especificación semántica de una base de datos de diferentes formas. Por el momento se proponen tres: generar un esquema Entidad-Relación, (re)generar un esquema Relacional, y hacer *browsing* de la base de datos tomando la especificación como guía.

**El módulo de acceso a la base de datos.** Este módulo resuelve el acceso a la base de datos desde los algoritmos de re-ingeniería y de explotación. También resuelve el acceso a datos correspondientes a objetos definidos en la especificación semántica.

**La interfaz al usuario.** Permite el acceso del usuario a los diferentes módulos, tanto de re-ingeniería como de explotación del resultado.



**Fig. 1 – Arquitectura de la herramienta DBRE.**

En lo relativo a la implementación, los módulos de construcción y de explotación de la especificación semántica están escritos en Prolog, mientras que la interfaz está desarrollada en Visual Basic. La Base de Conocimiento está implementada en Prolog, donde los elementos son átomos. El acceso a la base de datos está implementado como funciones C con SQL embebido que están linkeadas con los programas Prolog, y que estos llaman como predicados.

En las próximas secciones se presenta una descripción más detallada de la Base de Conocimiento, de los módulos de construcción de la especificación semántica, y de los de explotación de la misma.

### ***3.2 La Base de Conocimiento***

La herramienta de DBRE tiene por objetivo generar el modelo semántico de la base de datos sometida a estudio. Llevar a cabo esta tarea implica rastrear información usando las distintas familias de algoritmos, y obtener la mayor cantidad de información posible, por ejemplo claves primarias, claves foráneas, dependencias funcionales, relaciones, y el tipo de las mismas, etc. La base de conocimiento almacena la información generada durante el proceso global de re-ingeniería. Dicha información consiste en:

*La especificación semántica de la base de datos analizada:* información de los objetos de la base y las relaciones entre ellos.

*Aserciones de trabajo:* datos intermedios generados por los algoritmos de re-ingeniería, que constituyen la base para construir la especificación semántica de la base de datos.

## La especificación semántica de la BD

Consiste en una especificación de la semántica de la base de datos fuente representada a través de un Modelo Semántico. Dicho modelo representa un amplio espectro de relaciones entre los objetos de la base. La información representada consiste en *objetos* y *vínculos (links)*. Los objetos son *moléculas* (representaciones de objetos compuestos, e.g. entidades) o *átomos* (representaciones de objetos simples, e.g. atributos). Los *vínculos* representan relaciones definidas entre objetos, esquemáticamente:

	M	A
M	Gen_of Spec_of Part_of Composed_of Association	Has_attr
A	Attr_of	Is_part_of Comprises Sub_cat Super_cat

Vínculo	Inversa	Descripción
Gen_of(M1, M2, origen)	Spec_of	M1 es una generalización de M2.
Part_of(M1, M2, origen)	Composed_of	M1 es parte de M2.
Association(M1, M2, card, origen)		M1 está asociada a M2, con determinada cardinalidad.
Attr_of(A, M, m/mv, unicidad, origen)	Has_attr	A es atributo de M, de tipo mono o multivalorado.
Is_part_of(A1, A2, m/mv, origen)	Comprises	A1 es parte de A2, de tipo mono o multivalorado.
Sub_cat(A1, A2, nom_jerarq, origen)	Super_cat	A1 es subcategoría de A2, la jerarquía tiene nombre nom_jerarq.

Para cada vínculo detectado, se anota el *origen*: expresa la fuente de la cual se obtuvo la información, e.g. directamente del esquema de la base de datos, derivada de algoritmos, por intervención del usuario, etc.. Esta información resulta de interés para clasificar la calidad de la información obtenida;

## Las aserciones de trabajo

En las aserciones de trabajo se representa información del tipo de: dependencias funcionales, dependencias de inclusión, condiciones de join, caminos de join, atributos de group by, cardinalidad de las relaciones encontradas. A modo de ejemplo, se muestran las siguientes aserciones utilizadas cuando se utilizan los algoritmos de [PTB\*96].

Aserción	Descripción
r(NomTable)	Tablas de la base de datos.
a(Table, At, Unique?, Notnull?)	Atributos de las tablas y sus propiedades.
q(T1, Lat1, T2, Lat2)	Condición de join.
id(T1, Lat1, T2, Lat2, Confiabilidad)	Dependencia de inclusión y grado de confiabilidad de la información obtenida.
n(Table, Lat, Cant)	Conteo de tuplas según proyección de atributos.
njoin(T1, Lat1, T2, Lat2, Cant)	Conteo de tuplas resultantes de join.
k(Table, AttsKey) k(Table, [At]) :- a(Table, At, true, true)	Claves primarias.
s(Table)	Nuevas tablas derivadas.

### 3.3 Módulos de construcción de la especificación semántica

Los módulos que permiten la construcción de la especificación consisten en los algoritmos que realizan la re-ingeniería de la base de datos fuente y en el algoritmo que construye la especificación a partir de resultados intermedios generados por los primeros.

#### 3.3.1 Algoritmos de Re-ingeniería

En este módulo se realiza el núcleo del procesamiento de la re-ingeniería. Dentro del mismo se tienen diferentes familias de algoritmos que realizan procesamientos bien definidos. Estas familias son: (i) detectores de claves, (ii) detectores de dependencias funcionales y de inclusión, (iii) detectores de agrupamientos de atributos.

Dentro de cada familia se pueden tener diferentes algoritmos concretos que se basan en diferentes estrategias. Sin embargo, a pesar de tener algoritmos diferentes, todos los algoritmos de una familia generan información en un formato normalizado, esto es, dando valor a las mismas aserciones de trabajo. Por ejemplo, los algoritmos detectores de dependencias de inclusión generarán aserciones sobre este tipo de dependencias independientemente de cómo se realizó dicha detección.

Dentro de los *detectores de claves* existen algoritmos que se basan solo en información estructural, así como algoritmos que consultan los datos, y también algoritmos que analizan operaciones SQL sobre la base. En [PTB\*96] se detectan claves directamente del catálogo de la base de datos, mediante la búsqueda de atributos con declaraciones UNIQUE y NOT NULL. Por otra parte, en [And94] se analizan sentencias SQL de manipulación de los datos para confirmar o descartar atributos clave. Más concretamente, se deducen atributos clave a partir de sentencias SELECT INTO, y se descartan como clave los atributos en el WHERE de sentencias con DISTINCT en el SELECT.

Los algoritmos de la familia de *detectores de dependencias* (funcionales y de inclusión) se basan en la información de claves, y en condiciones de equi-Join encontradas en sentencias SQL [PTB\*96] [Chi95]. Luego comparan las cardinalidades de los atributos en cada condición realizando consultas a la base de datos. Observar que se están deduciendo restricciones de integridad a partir de una instancia particular de la base de datos. Por lo que es necesario que, como último paso el usuario confirme la corrección del conjunto de dependencias.

Los algoritmos de la familia de *detectores de agrupamientos de atributos*, detectan como se agrupan atributos en cláusulas GROUP BY. Este tipo de algoritmos no han sido considerados en las propuestas relevadas. Esto se debe, posiblemente, a que la mayoría de estas se orientan a la generación de esquemas Entidad-Relación y no a capturar la semántica de la base de datos en forma independiente a cómo se la desea representar. En particular, consideramos que la información sobre agrupamientos de atributos puede resultar muy útil para generar jerarquías en dimensiones de esquemas multidimensionales.

#### 3.3.2 Derivación de la especificación semántica

Este algoritmo toma como entrada a las aserciones de trabajo generadas por los algoritmos de re-ingeniería y construye la especificación del esquema semántico.

El algoritmo tiene en cuenta que la información en las aserciones puede haber sido generada por diferentes algoritmos y por lo tanto puede tener diferente confiabilidad. Por ejemplo, los resultados basados en información estructural son más confiables que los obtenidos a partir de consultas. Sin embargo, si solo se toma en cuenta este tipo de información, puede llegar a obtenerse una especificación

muy pobre. Para lograr un equilibrio, este algoritmo debe recibir indicaciones del usuario administrador sobre qué peso dar a la información generada en cada tipo de algoritmo.

### 3.4 Módulos de explotación de la especificación semántica

En la presente propuesta se separa netamente la construcción de la especificación semántica de su explotación. Mientras que la primera apunta a capturar toda la información posible sobre la base de datos fuente, la segunda (la explotación) genera una representación particular, en un modelo específico que se focaliza en ciertos constructores de la especificación semántica, así como provee mecanismos de tipo *browsing* para acceso a los datos guiados por la especificación semántica.

A continuación se presentan dos módulos diferentes de explotación: (i) generación de esquemas ER, y (ii) *browsing* de la base de datos.

#### 3.4.1 Generación de esquemas Entidad-Relación

Consiste en la generación de un esquema Entidad-Relación Extendido a partir de la especificación semántica. Esto no presenta dificultades ya que la especificación semántica modeliza todos los conceptos (y otros más) presentes en Modelos Entidad-Relación Extendido.

El algoritmo de generación tiene la forma de reglas Prolog que definen predicados que implementan el esquema Entidad-Relación en función de los predicados que representan la especificación semántica.

Las reglas son las siguientes:

```
entity_type(O) :- molecule(O),
                 not association(O,_,_,_).

relationship_type(O) :- molecule(O),
                       association(O,_,_,_).

link_Ent_Rel(R,E,Card,Tot) :- molecule(R),
                              molecule(E),
                              association(R,E,Card,Tot).

attributes(O,A,D,C,U) :- molecule(O),
                        atom(A,Domain),
                        has_attr(O,A,C,U).

specializacion(Super_E, Sub_E) :- molecule(Sub_E),
                                  molecule(Super_E),
                                  spec_of(Sub_E, Super_E).
```

#### 3.4.2 Browsing de la base de datos

Este es un programa que permite consultar la base de datos fuente a través de un mecanismo de navegación tipo *browser*. Para ello le presenta al usuario una representación gráfica de la especificación semántica obtenida de la base de datos fuente, y se le permite consultar los datos en las moléculas y átomos así como navegar por los links.

## 4 Conclusiones

En este artículo se presentó una herramienta para realizar Ingeniería Reversa de bases de datos relacionales. Esta herramienta apunta a construir una especificación semántica general que sea explotable



de diferentes formas, tanto bajo la forma de esquema conceptual (p. Ej., Entidad-Relación), como para realizar *browsing* de la base de datos.

Otra característica importante de la herramienta descrita es la integración de diferentes algoritmos de re-ingeniería. Tanto los existentes en la literatura como los aquí propuestos son, en general, complementarios. Esto es porque implementan diferentes estrategias de extracción de diferente tipo de información (claves, dependencias, etc). A modo de ejemplo, las estrategias que tienen en cuenta los datos de la base, además de la estructura y operaciones, son más potentes cuando hay una cantidad de datos suficiente para obtener conclusiones significativas; pero dan resultados muy poco confiables cuando las tablas están poco pobladas. Por lo tanto resulta útil contar con diferentes tipos de algoritmos aplicables en diferentes contextos. La condición para que esto sea operativo es que todos los algoritmos dentro de cada familia generen resultados en formato similar.

En lo relativo a los algoritmos de re-ingeniería, se implementaron varios de los descritos en la literatura. Además se propusieron otros para extraer agrupamientos de atributos usados en cálculos consolidados (*aggregates*).

### **Trabajo futuro**

El trabajo presentado en este artículo corresponde a resultados obtenidos en un proyecto que se encuentra en desarrollo. En el momento actual, se han especificado los algoritmos de construcción y explotación de la especificación semántica, y se están implementando los algoritmos de re-ingeniería y de derivación de la especificación semántica. Para los próximos meses se cuenta con completar una primer versión de la herramienta. Esta primer versión será usada en cursos sobre el tema para ser testeada y extendida a través de trabajos de estudiantes avanzados.

En cuanto a nuevos desarrollos, se ha comenzado a estudiar la generación de esquemas multidimensionales como resultado de la explotación de la especificación semántica. Para estos, la detección de agrupamientos de atributos resulta un aporte fundamental de información ya que da criterios para construir jerarquias en dimensiones. También se encuentra en estudio la conexión de la herramienta con metabases existentes. Esto tiene el objetivo de aplicar la herramienta propuesta en contextos donde ya se tienen descripciones de alto nivel sobre los objetos de las bases, pero no se cuenta con algoritmos que las construyan automáticamente.

---oOo---

## 5 Bibliografia

- [And94] M. Andersson. *Extracting an Entity Relationship Schema from a Relational Database Through Reverse Engineering*, Proc. 13<sup>th</sup> Int. Conf. on ER Approach, 1994.
- [CA97] I. Comyn-Wattiau, J. Akoka. *Reverse Engineering of Relational Database Physical Schemas*, 1997.
- [Che76] P. Chen. *The Entity Relationship Model – Toward a Unified View of Data*. ACM TODS 1(1), 1976.
- [Chi95] R. Chiang. *A Knowledge-based System for Performing Reverse Engineering of Relational Databases*, 1995.
- [HTJ\*94] J. Hainaut, C. Tonneau, M. Joris, M. Chandelon. *Transformation-based Database Reverse Engineering*, Proc. 12<sup>th</sup> Int. Conf. on ER Approach, 1994.
- [PTB\*96] J. Petit, F. Toumani, J. Boulicaut, J. Kouloumdjian. *Towards the Reverse Engineering of Denormalized Relational Databases*, Proc. of 12<sup>th</sup> Int. Conf. on Data Engineering, 1996.
- [RH97] S. Ramanathan, J. Hodges. *Extraction of Object-oriented Structures from Existing Relational Databases*, ACM SIGMOD Record Vol. 26 No. 1, 1997.

## Curriculum Vitae

**Raúl Ruggia** es Doctor en Informática (1996) de la Universidad Paris VI - Francia, Magister en Informática (1992) del PEDECIBA-Facultad de Ingeniería e Ingeniero de Sistemas en Computación (1990) de la Universidad de la República - Uruguay. Actualmente es Profesor Agregado del Instituto de Computación (InCo) de la Facultad de Ingeniería de la Universidad de la República, investigador del PEDECIBA y director del Laboratorio de Sistemas de Información de la Facultad de Informática. Sus áreas de interés son las bases de datos y las herramientas de ayuda al diseñador. Actualmente dirige proyectos de investigación en dichas áreas. Tiene numerosas publicaciones en conferencias regionales e internacionales. También trabaja como consultor en Sistemas de Información y Data Warehousing.

**Marcelo Colman** es Profesor de la Facultad de Informática del Instituto Universitario de Ciencias de la Información, Montevideo - Uruguay. Actualmente es investigador del Laboratorio de Sistemas de Información de la Facultad de Informática e integrante del Depto de Base de Datos e Ingeniería de Software de la Facultad de Informática. Sus áreas de interés son las bases de datos e Internet

**Gustavo Larriera** es Analista Programador (1986) de la Facultad de Ingeniería de la Universidad de la República – Uruguay. Actualmente es Profesor de la Facultad de Informática del Instituto Universitario de Ciencias de la Información, Montevideo – Uruguay, investigador del Laboratorio de Sistemas de Información de la Facultad de Informática y supervisor del Depto de Base de Datos e Ingeniería de Software de la Facultad de Informática. Sus áreas de interés son las bases de datos, Internet y la multimedia.