INFORMÁTICA

MÉTODOS DE INGENIERÍA INVERSA PARA BASES DE DATOS RELACIONALES

Resumen / ***Abstract***

La duración del ciclo de vida de un software depende esencialmente del rigor empleado en su implementación, así como del esfuerzo de mantenimiento desplegado para la organización que lo utiliza. No cabe ninguna duda del interés que porta la comunidad científica en cuanto al mantenimiento de estos, debido a las oportunidades brindadas para los nuevos paradigmas de desarrollo de software. La ingeniería inversa de bases de datos (DBRE), como propuesta, es un proceso imperfecto guiado por un conocimiento imperfecto. Así, numerosas técnicas y herramientas se basan en varias asunciones como precondición para su desarrollo. De esta forma resulta un proceso cuya aplicación es eficiente en ciertos sistemas de datos (SD) y no en otros. La reconstitución de estructuras semánticas dentro de las bases de datos relacionales (BDR) sigue siendo un asunto a investigar. Este trabajo presenta un marco de clasificación de algunas técnicas de DBRE relacionales ocurridas en los últimos años.

The life cycle sofware duration depends essentially on the efective effort used for its implementation and moreover for its maintenance. The underlooking on Database Reverse Engineering (DBRE) is an increasing fact-day, due to the opportunities provided by news paradigms of the software development industry. Database Reverse Engineering, as the aim, is an imperfect process driven by imperfect knowledge, so that many techniques and tools are based on some assumptions as the precondition of the purpose. Consequently, the result is just an effective and efficient method for some identified databases systems (DS). Inside the relational database (RDB), the semantic structure reconstruction is an actual task-investigation. Some recent algorithms were proposed, which don`t take any a-priori knowledge, and involve mapping techniques preserving this semantic information. This paper intends to give a classification of some relational DBRE occurred during lasts years.

Palabras clave / ***Key words***

DBRE, SD, información semántica, BDR, algoritmos, DBRE relacional

DBRE, DS, RDB, algorithms, semantic information, relational DBRE

Marc Desiré Atangana, *Ingeniero Informático, Centro de Estudios de Ingeniería de Sistemas (CEIS), Instituto Superior Politécnico José Antonio Echeverría, Cujae Ciudad de La Habana, Cuba*
e-mail:amarc@ceis.cujae.edu.cu
amarc.de@gmail.com

Roberto Sepúlveda Lima, *Ingeniero Electricista, Doctor en Ciencias Técnicas, Profesor Titular, Facultad de Ingeniería Industrial, Instituto Superior Politécnico José Antonio Echeverría, Cujae, Ciudad de La Habana, Cuba*
e-mail:sepul@ceis.cujae.edu.cu

Recibido:mayo del 2006
Aprobado:julio del 2006

INTRODUCCIÓN

La DBRE relacional consiste en un conjunto de técnicas que permiten construir una descripción conceptual correspondiente a una BDR, por ejemplo, un modelo **entidad-relación** (ERM).[1] Sus enfoques difieren según las motivaciones y los costos.[2] La DBRE relacional puede ser aplicada para resolver diferentes problemas, por ejemplo: reconstruir y(o) actualizar documentación perdida o inexistente de la BD, servir como pívot en un proceso de migración de datos, y ayudar en la exploración y extracción de datos en bases poco documentadas. Debido a la existencia de una cantidad importante de sistemas hechos con el modelo relacional, la DBRE relacional ha concitado la atención de la comunidad científica, dando a origen una cantidad importante de propuestas.

A pesar de sus diferencias, las propuestas realizadas tienen en común: el uso del ERM para representar el resultado del proceso, la orientación a resolver problemas de redocumentación de una BD, y la especificación de algoritmos puntuales más que la especificación de herramientas.[1]

REINGENIERÍA E INGENIERÍA INVERSA: SINOPSIS

En su papel titulado *Estado del arte de la reingeniería y la ingeniería inversa*,[3] en términos de porcentajes, presenta los trabajos realizados en ese campo. Se destaca así que los grupos más representativos han sido aplicaciones clásicas, aplicaciones Web, BD y sistemas no especificados (figura 1).

De hecho, las aplicaciones clásicas, de modo nativo, sistemas heredados más antiguos, más difundidos y, aun más utilizados hoy en día, constituyen el grupo de primera importancia, pues constituyen el 59 % de las investigaciones. Siguen las BD, como sistemas heredados estudiados con más frecuencia (18%),[3] En este caso, los trabajos son de dos tipos, o bien se trata de la recuperación eficiente de la estructura de la base de datos a partir de los ficheros existentes; o bien, de obtener una representación a más alto nivel (modelos conceptuales) a partir de un esquema relacional. En este último caso, se intenta obtener modelos que representen lo más fielmente la semántica inicial, recurriendo al uso de heurísticas y otras técnicas.

Las aplicaciones Web, son bastante recientes, aunque gozan tan solo del 10 % de los trabajos seleccionados, cabe destacar que prácticamente el 100 % de los mismos están orientados en la ingeniería inversa. Esto se debe a que la demanda actual de sistemas de este tipo provoca que se realicen ciclos de vida cortos para atender a las demandas del mercado.

En la referencia 3 se estipula que la ingeniería inversa ocupa más del 80 % del volumen total en cuestión de investigación y publicaciones, dejando menos de un 20 % de los trabajos para la reingeniería del software.

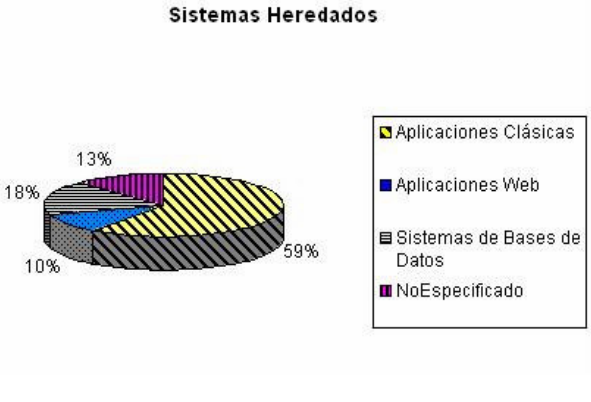

Fig. 1 *Proporción de los tipos de sistemas heredados.*

Una de las razones por las que puede ocurrir esto, es que en todo proceso de reingeniería, siempre hay una etapa de ingeniería inversa, mientras que la ingeniería inversa puede llevarse a cabo de manera separada, sin que se pretenda emprender a posteriori una etapa de ingeniería directa.

La tarea relacionada con la ingeniería inversa suele ser compleja y pesada debido a que en muchos casos, el substrato matemático en el que están basadas algunas de estas técnicas hace que su aplicación manual sea prácticamente inabordable. Por eso, son muchos los autores que proponen alguna herramienta que ofrezca soporte al método del que versa su publicación. El disponer de una herramienta o familia de herramientas que automatice, de forma total o parcial, el proceso de recuperación de las especificaciones abstractas, revoca en un ahorro importante de tiempo y dinero, además de facilitar la comprensión del mismo mediante la posibilidad de visualizar el sistema desde distintos puntos de vista.[3]

MARCO DE CLASIFICACIÓN

La estructura orgánica de una base de datos relacional está compuesta de tablas interrelacionadas según una semántica bien definida. A veces esa BD no ha sido completamente normalizada o ha sido modificada luego de su creación sin tener en cuenta las restricciones definidas en su DDL; también puede ser que la pluralidad de consultas no respecta siempre las restricciones de integridad. A consecuencia de esos factores, la BD se encuentra entonces inestable.

Hoy en día, existen una cantidad de propuestas. Cada una exhibe sus propias características de procesamiento, produce sus propias salidas, y requiere específicas entradas y asunciones. Así que numerosas propuestas se enfatizan en primer lugar en el análisis de las relaciones entre tablas para comprender la semántica actual de la BD. Juntos a la cosecha de la documentación aún disponible y al estudio del entorno global del sistema, estos pasos constituyen el inicio de la DBRE. El mecanismo de clasificación presentado esta basado, en esencia, en las asunciones y requisitos de entradas, los resultados, así como la factibilidad de cada método.

Métodos

A continuación se describe cada método, organizándolo por categoría:

- **Entrada:** Tipos de información que se requiere para emprender el método.
- **Asunciones**: Precondiciones que hace el autor en el alcance de su método.
- **Salida:** Resultados obtenidos.
- **Metodología:** Etapas mayores seguidas.
- **Factibilidad:** Particularidad propia de la propuesta.

Método de Chiang et al (1995)

Entrada: Este método requiere como entrada, las instancias de datos y los esquemas de relación, incluyendo claves primarias.

Opcionalmente, el usuario podría insertar algunas dependencias de inclusión.

Asunciones: Se supone las relaciones en tercera forma normal (3NF), la consistencia de nombres de los atributos, la ausencia de errores en los valores de los atributos.

Salida: Un modelo entidad-relación extendido (EER).

Metodología: La propuesta se enfoca en tres pasos esenciales, a) la clasificación de las relaciones y atributos, basándose en los esquemas de relaciones y sus llaves primarias; b) la generación/ verificación de dependencias de inclusión haciendo uso de heurísticas basados en los atributos ordenados, contra las instancias de datos; c) la identificación de los componentes del EER, usando una lista de reglas definidas.

Factibilidad: La propuesta de Chiang se fundamenta en la generación de dependencias de inclusión y la completa justificación de las transformaciones aplicadas a la BD para obtener el EER resultante.[4,5]

Método de Petit et al (1996)

Entrada: Este método requiere como entrada los esquemas de relación, las relaciones con restricción de unicidad no nulos, las instancias de datos y el código.

Asunciones: Ninguna.

Salida: Un Modelo entidad-relación extendido (EER).

Metodología: En primer lugar, se busca las dependencias de inclusión haciendo uso de esquemas de relación, instancias de BD y las consultas de *equi-join* en el código. Luego el método busca las dependencias funcionales pertinentes entre atributos no llaves, usando esquemas de relaciones, llaves candidatas, las dependencias de inclusión no rechazadas por el usuario y el algoritmos de descomposición-normalización en 3NF. Así que se hace un mapeo del esquema relacional para obtener un EER.

Factibilidad: El método de Petit *et al*. se distingue por que incluye una normalización del esquema relacional. Sin embargo, ciertos objetos voluntariamente escondidos por el usuario, o no revelados por las consultas de *equi-join* pudieran permanecer.[4,5]

Método de Premerlani y Blaha (1998)

Entrada: Este método requiere como entrada los esquemas de relación y los datos.

Asunciones: Ninguna.

Salida: Se obtiene un modelo OMT.

Metodología: En primer lugar, se prepara un modelo objeto inicial, considerando cada tabla como una clase posible. Haciendo uso de esas premisas, el usuario podrá utilizar las llaves primarias como identificadores de clases, también podrá determinar los grupos de llaves ajenas. Gradualmente, se hace el refinamiento del esquema OMT basándose en la guía en línea que incluye las consultas de datos.

Factibilidad: El método de Premerlani y Blaha esta caracterizado que provee solo pocos pasos mecánicos, debido a un grado elevado de interacción del usuario. Intenta por procesar algunas delicadas representaciones, dando algunas directivas de optimización de diseño.[4]

Método de Pedro Sousa et al (1999)

Entrada: Se requiere como entrada el modelo lógico, el código y los datos.

Asunciones: Relaciones en 3NF.

Salida: Se obtiene un esquema conceptual refinado.

Metodología: La propuesta de Pedro Sousa *et al* se articula en tres fases esenciales.

La fase 1, nombrada *Agrupación de tablas* abarca pasos tales la identificación de claves primarias, la agrupación de tablas en entidades abstractas e interrelaciones. La fase 2 proporciona un refinamiento de entidades abstractas obtenidas precedentemente. En la fase 3, se trata de integrar los diferentes esquemas conceptuales intermedios en un esquema conceptual global.

Factibilidad: El método de Pedro Sousa se distingue por que simplifica el proceso de comprensión del dominio de la aplicación, permite el uso de otros métodos de DBRE (Petit *et al*, Chiang *et al*) en el procesamiento de entidades abstractas.[5]

Método SOT (2001)

Entrada: Este método requiere como entrada el esquema de relaciones.

Asunciones: Ninguna.

Salida: Se obtiene un modelo orientado a objeto.

Metodología: El método de Behm *et al* introduce un nuevo modelo de dato llamado SOT (*Semi Object Types*) o tipo semi-objeto que se fundamenta en el álgebra matemática. El proceso de transformación de esquema esta dividido en tres tareas secuenciales y los datos del proceso de migración están generados automáticamente :

- La transformación del esquema relacional en un esquema SOT.
- El rediseño del esquema SOT.
- El mapeo del esquema SOT en un esquema orientado a objeto.

Factibilidad: El modelo de dato SOT introducido por *Behm et al* utiliza técnicas algebraica para reconstruir la arquitectura semántica del esquema relacional inicial. Luego el mapeo del esquema SOT hacia un esquema orientado a objeto se hace a través de una interfaz de notación de ODMG ODL (Object Definition Language).[6]

Método UQoRE (2001)

Entrada: El método requiere como entrada una base de consultas de usuarios, además de todas las fuentes de información habituales tales esquemas de relación, programas de aplicaciones.

Asunciones: Ninguna.

Salida: Se obtiene un modelo conceptual orientado a objeto.

Metodología: El método de Azíz Barbar se fundamenta en la explotación de consultas usuarios pues considerando estas de valiosa riqueza semántica. Se articula en los puntos siguientes:

- *Extracción de similaridad*: En primer lugar, se trata de dar un nombre único a cada atributo de forma tal que se vuelva

consistente el nombramiento de los atributos que tienen el mismo concepto. Al mismo tiempo, se identifica los atributos de conceptos diferentes.

• *De optimización*: Esta etapa incluye la normalización y la reestructuración del esquema. Se obtiene un esquema en 3NF donde están detectadas dependencias funcionales y escindidas las relaciones que no estaban en 3NF. La reestructuración consiste en detectar estructuras candidatos distintos y similar para combinarlas o separarlas, y obliterar demás.

• *Descubrimiento de claves primarias /ajenas*: Aquí se buscan enlaces (claves primarias) de generalización / especialización, y otros enlaces entre clases estarán mostrados por las claves ajenas. El descubrimiento de dependencias de inclusión será también hecho entre los diferentes componentes del esquema de la base de datos.

• *Translación de esquema*: Una vez que la BD esta en 3NF, se transforma cada relación en una clase o una asociación entre clases según la estructura de la clave primaria. Los enlaces de generalización /especialización se mostrarán también.

Factibilidad: La particularidad del método UQoRE (*User Query Oriented Reverse Engineering*) está en el uso de técnicas de minería de datos desde una base de consultas usuarios que habrían sido almacenadas. El resultado es un modelo conceptual objeto estático.[7]

Método de Hainaut et al (1994-2003)

Entrada: Este método requiere como entrada los esquemas de relación, el código y cualquier otra fuente de datos (documentación, reportes, pantallas, otros).

Asunciones: Ninguna.

Salida: Se obtiene un esquema conceptual normalizado que puede ser transformado en un modelo propio a otra paradigma (orientado a objeto).

Metodología: La metodología esta basada en un enfoque transformacional. Abarca dos fases principales, la extracción de estructura de datos y la conceptuación de estructura de datos. La fase 1 comprende los puntos siguientes:

• Analizar el esquema de relaciones y el código DMS-DDL.

• Integrar los esquemas físicos parciales obtenidos anteriormente.

• Refinar el esquema para recuperar las construcciones implícitas.

La fase 2 se articula en dos puntos, la conceptuación básica (eliminación de estructuras de optimización) y la normalización conceptual.

Factibilidad: Hainaut *et al*., propone una metodología genérica y secuencial de DBRE que se podría especializar en varios modelos de datos en los cuales se fundamenta la mayor parte de sistemas heredados, tal los ficheros estándares COBOL, los modelos CODASYL, IMS, las BD relaciónales y el modelo objeto.[8,9] Ver figura 2.

Método de Jahnke et al (2002)

Entrada: Este método requiere como entrada los esquemas de relación, el código y cualquier otra fuente de datos (documentación, reportes, pantallas, otros).

Asunciones: Ninguna.

Salida: Se obtiene un esquema conceptual orientado a objeto.

Metodología: La propuesta de Jahnke *et al* es un proceso iterativo y explorativo de DBRE abarcando tres pasos: a) el análisis del esquema lógico, por fin de asegurar un enriquecimiento semántico con las nuevas informaciones proporcionadas para el código y la documentación colectada; b) la transformación del esquema lógico enriquecido mediante técnicas de la gramar de grafos. En primer lugar, se obtiene un esquema conceptual relacional, y por ende , c) el rediseño de este esquema conceptual, añadiendo nuevas funcionalidades. Mediante técnicas de grafos de mapeo, este esquema será transformado en un esquema conceptual orientado a objeto.

Factibilidad: En comparación con otros procesos de DBRE, el método de Jahnke *et al.* proporciona un proceso iterativo y explorativo centrado-usuario, basándose en las observaciones descritas para Premerlani y Blaha.[10] Ver figura 3.

TABLA DE CLASIFICACIÓN

En la tabla de clasificación (tabla 1), se muestra un resumen de los métodos descritos anteriormente.

ANÁLISIS COMPARATIVO

En la literatura abundan algoritmos para las tareas de ingeniería inversa como la transformación canónica de esquemas de la BD.[10] Sin embargo, existen también varias críticas por parte de los mismos autores, en cuanto a las limitaciones de las propuestas de los demás. Autores como Premerlani y Blaha, Aziz Barbar, Petit *et al*, Chiang y *et al*., enfatizan que un enfoque interactivo, centrado-usuario tiene más éxito plausible que un proceso orientado-lote con compiladores. No obstante, ellos no proporcionan ningún soporte para esta interactividad hombre-maquina.[10]

Behm *et al*, propone un entorno interactivo de migración de esquema que provee un conjunto de reglas de mapeo. En cada etapa, el usuario deberá escoger adecuadamente la regla de mapeo para cada artefacto del esquema. Hainaut y *et al.*, despliega un modelo de datos genérico común que abarca reconstrucciones de modelo conceptual, así como modelo lógico (y físico). Basándose en ese modelo de dato, Hainaut *et al.,* han definido un catalogo de transformaciones de esquemas que restituyen de forma gradual construcciones implícitas a nivel bajo de implementación, mediante varios conceptos abstractos.[9] Sin embargo, la ejecución de este enfoque impide un proceso de DBRE iterativo, pues la estructura (lógica) original del esquema se pierde durante el proceso de transformación.[10]

Para evitar esa pérdida de información, algunos autores tales Jeusfeld y Johnen han propuesto el uso de un metamodelo genérico como mediator.[11] La propuesta de Jahnke *et al.*, análoga a la de Premerlani y Blaha a sus emprendimientos, provee también un catalogo de mapeo de esquemas, centrándose en una interacción con el usuario, a fin de garantizar la máxima conservación de la información semántica contenida en los diferentes artefactos de esquemas,[10] aunque la flexibilidad de esa propuesta no siempre es evidente.

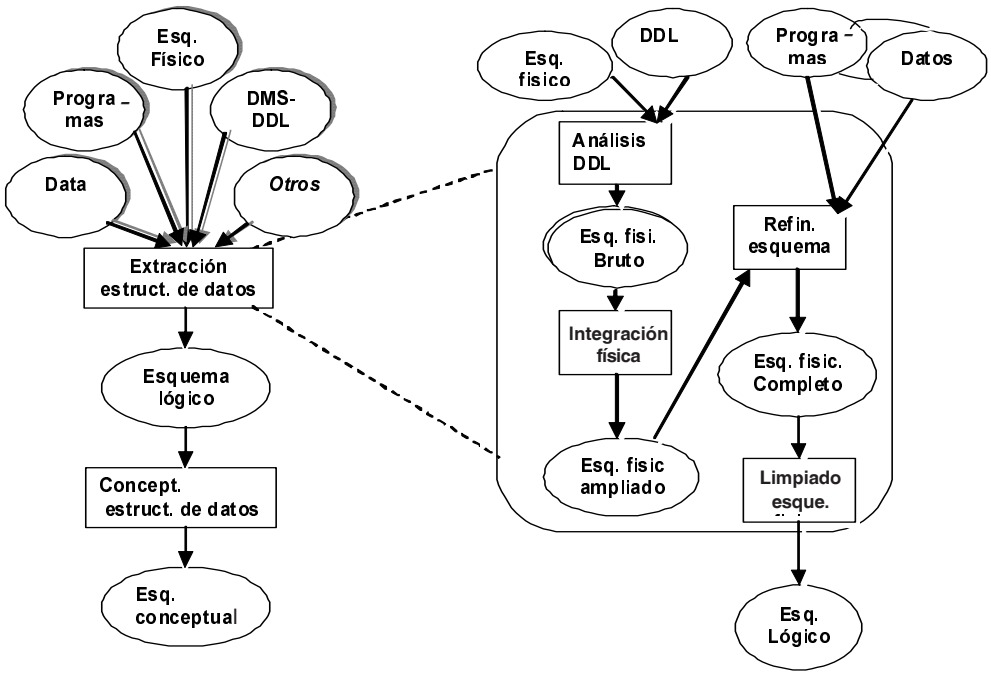

Fig: 2 *El proceso mayor de la metodología de referencia DBRE (izquierda) y el desarrollo del proceso de extracción de estructuras de datos (derecha).*

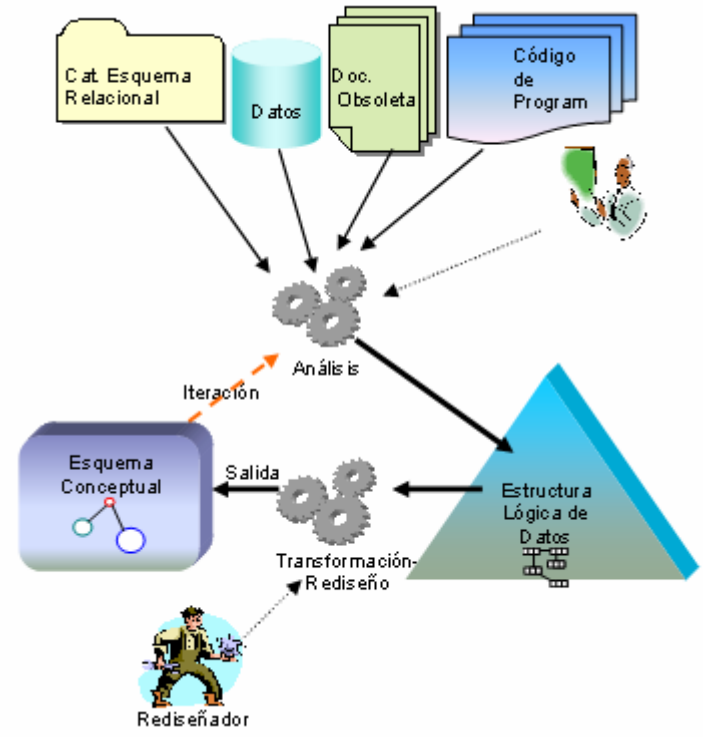

Fig. 3 *Catálogo del proceso de reingeniería según el método de Jahnke et al.*

TABLA 1
Resumen de los métodos descritos

Método	Entrada	Salida	Precondiciones	Tiene herramienta
Chiang *et al*	Datos Relaciones Claves primarias	Modelo EER	3NF Consistencia de nombres Ausencia de errores en las claves primarias	No
Petit *et al*	Relaciones con restricciones de unicidad no nulas Datos Código	Modelo EER	Ninguna	No
Premerlani y Blaha	Relaciones Datos	Modelo de clases OMT	Ninguna	No
Pedro Sousa *et al*	Modelo lógico Código; Datos	Modelo conceptual relacional	Relación en 3NF	No
SOT	Esquema relacional	Modelo orientado a objeto	Ninguna	Sí
UqoRE	Base de consultas usuarios; Relaciones Código; Datos	Modelo orientado a objeto	Ninguna	No
Hainaut *et al*	Relaciones Código; Datos	Esquema conceptual normalizado	Ninguna	Sí
Jahnke *et al*	Relaciones Código ; Datos	Modelo conceptual orientado a objeto	Ninguna	Sí

A pesar del esfuerzo de conjunto apreciable, realizado por los autores, hoy en día se notará que numerosas herramientas soporte de DBRE aún están en prueba y su desarrollo-refinamiento sigue siendo una tarea priorizada para sus autores respectivos. El debate en cuanto a la eficacia de un método en comparación con otros permanece siendo actual, mientras continúan las investigaciones en el entorno de DBRE.

CONCLUSIONES

No cabe ninguna duda que el modelo relacional sigue siendo uno de los más utilizados hoy en día, dadas las necesidades empresariales, las investigaciones en cuanto a su mantenimiento se producen con un esfuerzo creciente.

En este trabajo, en el que se han presentado algunos de los métodos más recientes de DBRE relacional, se destaca un neto desempeño de cambio de paradigma (del relacional hacia el orientado a objeto). Las motivaciones son numerosas, las técnicas variables, pero la finalidad es la misma. También cabe destacar que, en un gran porcentaje de los estudios, se proponen (bien existentes en el mercado, bien de propio desarrollo, y en mejoramiento constante) herramientas que ofrecen soporte a la propuesta teórica,[3] que exponen los autores en sus publicaciones. A pesar de la ausencia de un modelo de DBRE estándar, la comprensión del mecanismo de las propuestas es de vital importancia, para escoger y profundizar una de ellas. Así, queda claro que los resultados aquí presentados deben ser ampliados con un estudio a mayor escala.

REFERENCIAS

1. **COLMAN, M.; G. L. AND R. RUGGIA:** *Database Reverse Engineering : Proposal an Open Tool Based on Semantic Model*, Laboratorio de Sistemas de Información, Facultad de

Informática del instituto Universitario de Ciencias de la Información, 1997.

2. **BARBAR, A. M. C.:** *Semantic Extraction: A User-Driven Method*, I3S Laboratory, University of Nice-France, 2000.
3. **GARCIA, I.** *et al.*: *Estado del arte de la reingeniería y la ingeniería inversa : 2001-2003. 2004*: Grupo ALARCOS, Departamento Informática, Universidad de Castilla-La Mancha-España, 2004.
4. **PEDRO, LOURDES DE JESÚS:** *Selection of Reverse Engineering Methods for Relational Databases*, IST/INESC, Lisboa Codex-Portugal, 1999.
5. **RUIZ, F. Y M. P.:** *Mantenimiento de Software*, Departamento de Informática, Universidad de Castilla-La Mancha, Escuela Superior de Informática, Ciudad Real, 2001.
6. **BEHM, A.:** *Migrating Relational Databases to Object Technology*, Department of Computer Science, University of Zurich, May 16, 2001.
7. **BARBAR, A.:** *User Driven Method for Database Reverse Engineering*, I3S Laboratory, University of Nice- France,2001.
8. **HENRARD, J.** *et al.*: *Database Reegineering*, Facultés Universitaires de Namur, Institut d'informatique, LIBD-Laboratoire d'ingénierie des Applications de Bases de Donnees June 25, 2003.
9. **HAINAUT, J. L.:** *Contribution to the Reverse Engineering of OO Applications-Methodology and Case Study*, Institut d'informatique, University of Namur. Belgium, 2003.
10. **JENS, H.; W. S., JAHNKE; JORG WADSACK AND ALBERT ZUNDORF**: *Supporting Iterations in Exploratory Database Reengineering Processes*, Department of Computer Science, Univ. of Victoria-Canada. Software Engineering Group, Dept. of Mathematics and Computer Science, Univ. of Paderborn. Dept. of Maths and Computer Science, Technical Univ. of Braunschweig, 2002.
11. **JEUSFELD, A.** *et al.*: *An Executable Meta Model for Re-Engineering of Database Schemas*, Informatik V, RWTH Aachen- Germany, 1994.

II Taller de Informática Aplicada

Presidente del Comité Organizador:
Dr. Ing. Miguel Garay
e-mail:garay@ceis.cujae.edu.cu

Este tendrá lugar en el marco del
IV Simposio de Ingeniería Industrial, Informática y Afines

Del 28 de noviembre al 1 de diciembre del 2006