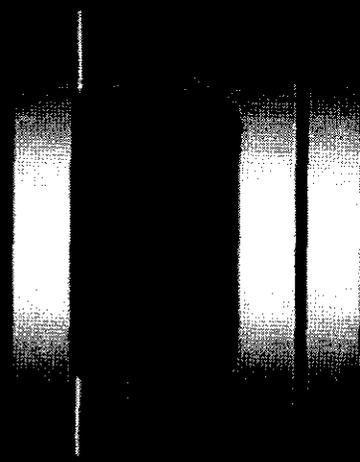


I+D Computación

Una publicación de la Academia de Posgrado de Ciencias Computacionales



Marzo de 2003. Vol. 2 No. 1

ISSN 1665-238X

Calidad de Modelos Lógicos de Bases de Datos

Coral Calero, Mario Piattini, Marcela Genero
Grupo de Investigación Alarcos, E. S. Informática,
Paseo de la Universidad, 4, 13071 Ciudad Real (España)
{Coral.Calero, Mario.Piattini, Marcela.Genero}@uclm.es

Abstract.– We devote this paper to present a state of the art on metrics for quality assurance (in particular maintainability) of database logical models. The study will center on relational, active and object-relational databases.

Keywords: database quality, conceptual modeling, databases.

1. INTRODUCCIÓN

Existen centenares de métricas que han sido propuestas a lo largo de los años, sin embargo, las métricas para bases de datos han sido descuidadas (Sneed y Foshag, 1998) y creemos que, debido a la gran importancia que las bases de datos han adquirido, resulta ineludible acabar con este vacío.

La inexistencia de métricas específicas para bases de datos puede ser explicada ya que hasta hace no demasiado tiempo, las bases de datos jugaban un papel relativamente secundario dentro de los sistemas de información siendo los programas los verdaderos protagonistas. Ello justifica la gran presencia de métricas orientadas a programas que podemos encontrar en la literatura, propuestas por muy diferentes autores. Sin embargo, aunque las bases de datos se han convertido en el corazón de los Sistemas de Información (SI) más relevantes para el funcionamiento de la sociedad, su diseño sigue siendo una tarea larga, difícil y costosa (De Miguel, Piattini y Marcos, 1999). Además, el tamaño y la naturaleza de los datos pueden influir, en gran medida, en muchos aspectos de un SI como el esfuerzo de desarrollo (MacDonnell et al., 1997).

Así pues, consideramos de primordial importancia la definición de métricas para asegurar la calidad de las bases de datos. La calidad, según la ISO 9126 (ISO, 1999) está afectada por varios factores. Uno de estos factores es la mantenibilidad¹, factor especialmente delicado ya que el mantenimiento representa el mayor problema del desarrollo software suponiendo entre el 60 y el 80 por ciento de los costes del ciclo de vida (Card y Glass, 1990; Pigoski, 1997).

Según la ISO 9126, tres de los factores que influyen en la mantenibilidad son la analizabilidad², la cambiabilidad y la facilidad de prueba, los cuales a su vez, según Li y Cheng (1987) están influenciados por la complejidad. Henderson-Sellers (1996) divide la complejidad en tres: complejidad computacional, complejidad psicológica y complejidad representacional, y para la psicológica define tres componentes:

¹ Aunque “maintainability” debería traducirse correctamente por “facilidad de mantenimiento”, nosotros utilizaremos el término “mantenibilidad” por ser el que generalmente se utiliza.

² La analizabilidad de la ISO 9126 es equivalente a la comprensibilidad de Li y Chen (1987) y a la entendibilidad para nosotros.

complejidad del problema, factores cognitivos humanos y complejidad del producto. A su vez, las métricas de producto pueden ser divididas en intra-modulares e intermodulares (Kitchenham y Linkman, 1990). En la complejidad del producto, tanto intermodular como intramodular es en la que se ha centrado nuestro trabajo.

Por tanto, el objetivo fundamental será diseñar métricas de producto (que capturen las características específicas de los diferentes tipos de bases de datos) que nos permitan medir la complejidad de las bases de datos, a través de lo cual estaremos también midiendo la calidad de las mismas.

Concretamente, centraremos el estudio en las bases de datos relacionales, activas y objeto-relacionales³, ya que:

- Las bases de datos relacionales son las más implantadas en la actualidad (Leavitt, 2000).
- Los productos comerciales incluyen algún tipo de actividad y, sin embargo, los disparadores presentan grandes problemas de mantenimiento (Kotz, 1994).
- Las bases de datos objeto-relacionales se están imponiendo como las bases de datos del futuro, ya que incluyen todas las ventajas de las bases de datos relacionales y la potencia de las orientadas a objetos. Teniendo en cuenta que hay estudios que aseguran que en el año 2003 las bases de datos objeto-relacionales sustituirán a las relacionales (Leavitt, 2000), consideramos imprescindible tener métricas que nos permitan asegurar su calidad.

Vamos a utilizar el método GQM (Basili y Weiss, 1984; Rombach, 1990) para presentar las métricas. Siguiendo este método, para cada tipo de base de datos, los pasos a seguir son:

- Definir claramente el objetivo que queremos alcanzar mediante la definición de métricas.
- Preguntarse cómo, a través de las características específicas del producto a medir, podemos alcanzar nuestro objetivo.
- Intentar responder a cada una de esas preguntas mediante la definición de métricas

En el resto de este artículo presentamos las métricas definidas para bases de datos relacionales en la sección 2. En la sección 3 explicamos las métricas para bases de datos activas y las objeto-relacionales en la sección 4. Por último, en la sección 5 presentamos las conclusiones de nuestro trabajo.

2. BASES DE DATOS RELACIONALES

Date (1995) define un sistema de gestión de base de datos relacional como “un sistema, en el que como mínimo:

- Los datos son vistos por el usuario como tablas (y sólo así), y
- Los operadores disponibles para el usuario generan nuevas tablas a partir de otras antiguas. Entre los operadores se encuentran, como mínimo la selección, (SELECT), la proyección (PROJECT) y la combinación (JOIN)”.

³ Aunque en principio, tal y como señala Stonebraker y Brown (1999) las bases de datos objeto-relacionales abarcan las relacionales, las activas y las extensiones objetuales, nosotros nos referiremos como objeto-relacionales sólo a las del último tipo, por motivos de claridad en la presentación.

A continuación explicaremos las métricas utilizando GQM y posteriormente, presentaremos un ejemplo de aplicación de las métricas.

- **Objetivo (Goal)**

Propósito: Asegurar la
 Asunto: mantenibilidad
 Objeto: de las bases de datos relacionales
 Punto de vista: desde el punto de vista del diseñador de la base de datos

- **Preguntas (Question)**

1. ¿Cómo influye la complejidad de las tablas en la mantenibilidad de las bases de datos relacionales?
2. ¿Cómo influye la complejidad entre tablas en la mantenibilidad de las bases de datos relacionales?

- **Métricas (Metric)**

Para responder a la primera pregunta proponemos las siguientes métricas:

Número de Atributos de una Tabla (NA(T), *Number of Attributes*).

Definida como el número de atributos de una tabla T.

Número de Claves Ajenas (NFK(T), *Number of Foreign Keys*).

Definida como el número de claves ajenas de una tabla T.

Profundidad del Árbol Referencial de una Tabla (DRT(T), *Depth of the Referential Tree*).

Mediante la métrica DRT(T), medimos la profundidad máxima de todos los caminos referenciales del grafo que se forma, si tomamos la tabla T como el nodo raíz del grafo y todas las tablas relacionadas con T mediante integridad referencial como el resto de nodos y siendo las relaciones de integridad referencial los arcos del mismo.

Ratio de Claves Ajenas de una Tabla (RFK(T), *Ratio of Foreign Key*).

Definida como el porcentaje de atributos de la tabla T que son claves ajenas.

$$RFK(T) = \frac{NFK(T)}{NA(T)} \quad \text{donde:} \quad \begin{array}{l} NFK(T) \text{ es el número de claves ajenas de la tabla} \\ NA(T) \text{ es el número de atributos de la tabla} \end{array} \quad (1)$$

Y las métricas para responder a la segunda pregunta son:

Número de Tablas (NT, *Number of Tables*).

Definida como el número total de tablas que hay en el esquema.

Cohesión del Esquema (COS, *Cohesion of the Schema*).

Definida como la suma del número de tablas al cuadrado que hay en cada componente no conexas del grafo del esquema, siendo los nodos de este grafo las tablas del esquema y los arcos las relaciones de integridad referencial. Basándonos en lo anterior, esta métrica se define mediante la siguiente fórmula:

$$COS = \sum_{i=1}^{|US|} NT_{USi} \quad \text{donde:} \quad \begin{array}{l} |US| \text{ es el número de subgrafos no relacionados.} \\ NT_{USi} \text{ es el número de tablas en el subgrafo no} \\ \text{relacionado } i. \end{array} \quad (2)$$

Ratio de Normalidad (NR, Normality Ratio).

Definida como el número de tablas en tercera forma normal (o superior) entre el número total de tablas. La fórmula que formaliza la definición dada es:

$$NR = \frac{NT3NF}{NT} \quad \text{donde:} \quad \begin{array}{l} NT3NF \text{ es el número de tablas en } 3NF \text{ (Third} \\ \text{Normal Form) o superior} \\ NT \text{ es el número de tablas en el esquema} \end{array} \quad (3)$$

Pero también podemos utilizar algunas de las propuestas para tablas, pero orientadas al esquema:

Número de Atributos (NA, Number of Attributes).

Definida como el número total de atributos que hay en el esquema.

$$NA = \sum_{i=1}^{NT} NA(T_i) \quad (4)$$

Número de Claves Ajenas (NFK, Number of Foreign Keys).

Definida como el número total de claves ajenas que hay definidas en el esquema.

$$NFK = \sum_{i=1}^{NT} NFK(T_i) \quad (5)$$

Profundidad del Árbol Referencial (DRT, Depth of the Referential Tree).

Mediante la métrica DRT, medimos la profundidad máxima de todos los caminos referenciales del grafo que se forma si tomamos las tablas del esquema como los nodos y las relaciones de integridad referencial como los arcos del mismo.

$$DRT = \max_{i=1}^{NT} (DRT(T_i)) \quad (6)$$

Ratio de Claves Ajenas (RFK, Ratio of Foreign Key).

Definida como el porcentaje de atributos del esquema que son claves ajenas.

$$RFK = \frac{NFK}{NA} \quad \text{donde:} \quad \begin{array}{l} NFK \text{ es el número de claves ajenas del esquema} \\ NA \text{ es el número de atributos del esquema} \end{array} \quad (7)$$

Ejemplo

A continuación, en la Tabla 1, presentamos una base de datos relacional tomada de Elmasri y Navathe (1997):

Tabla 1. Ejemplo

<pre> CREATE TABLE EMPLEADO (NOMBREP VARCHAR(15) NOT NULL, INIC CHAR, APELLIDO VARCHAR(15) NOT NULL, NSS CHAR(9) NOT NULL, FECHAEN DATE, DIRECCION VARCHAR(30), SEXO CHAR, SALARIO DECIMAL(10,2), NSSUPER CHAR(9), ND INT NOT NULL, CONSTRAINT CLPEMP PRIMARY KEY (NSS), CONSTRAINT CLESUPEREMP FOREIGN KEY (NSSUPER) REFERENCES EMPLEADO(NSS) ON DELETE SET NULL ON UPDATE CASCADE, CONSTRAINT CLEDEPTOEMP FOREIGN KEY (ND) REFERENCES DEPARTAMENTO (NUMEROD) ON DELETE SET DEFAULT ON UPDATE CASCADE); CREATE TABLE DEPARTAMENTO (NOMBRED VARCHAR(15) NOT NULL, NUMEROD INT NOT NULL, NSSGTE CHAR(9) NOT NULL, FECHAINICGTE DATE, CONSTRAINT CLPDEPTO PRIMARY KEY (NUMEROD), CONSTRAINT CLSDEPTO UNIQUE(NOMBRED), CONSTRAINT CLEGTDEPTO FOREIGN KEY (NSSGTE) REFERENCES EMPLEADO(NSS) ON DELETE SET DEFAULT ON UPDATE CASCADE); </pre>	<pre> CREATE TABLE LUGAR_DEPTS (NUMEROD INT NOT NULL, LUGARD VARCHAR(15) NOT NULL, PRIMARY KEY (NUMEROD, LUGARD), FOREIGN KEY (NUMEROD) REFERENCES DEPARTAMENTO (NUMEROD) ON DELETE CASCADE ON UPDATE CASCADE); CREATE TABLE PROYECTO (NOMBREPR VARCHAR(15) NOT NULL, NUMEROPR INT NOT NULL, LUGARPR VARCHAR(15), NUMD INT NOT NULL, PRIMARY KEY (NUMEROPR), UNIQUE (NOMBREPR), FOREIGN KEY (NUMD) REFERENCES DEPARTAMENTO (NUMEROD)); CREATE TABLE TRABAJA_EN (NSSE CHAR(9) NOT NULL, NUMP INT NOT NULL, HORAS DECIMAL(3,1) NOT NULL, PRIMARY KEY (NSSE, NUMP), FOREIGN KEY (NSSE) REFERENCES EMPLEADO (NSS), FOREIGN KEY (NUMP) REFERENCES PROYECTO (NUMEROPR)); CREATE TABLE DEPENDIENTE (NSSE CHAR(9) NOT NULL, NOMBRE_DEPEND VARCHAR(15) NOT NULL, SEXO CHAR, FECHAAN DATE, RELACION VARCHAR(8), PRIMARY KEY (NSSE, NOMBRE_DEPEND), FOREIGN KEY (NSSE) REFERENCES EMPLEADO (NSS)); </pre>
--	--

En la Figura 1 se presenta el grafo relacional para el ejemplo anterior.

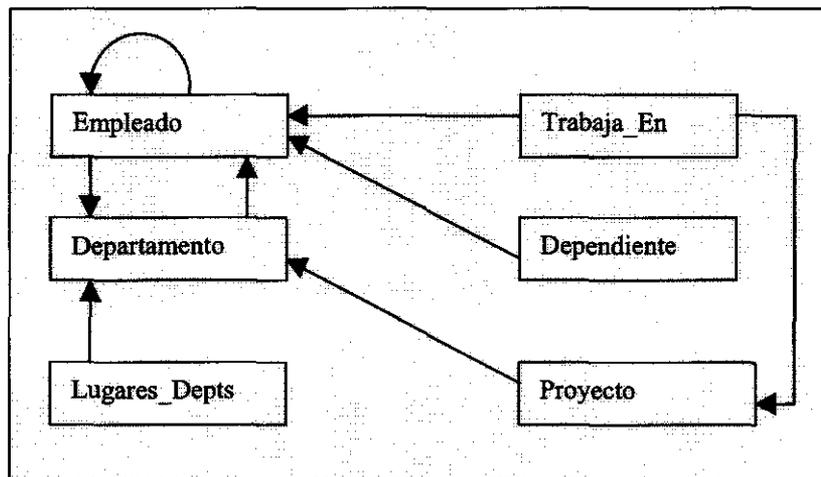


Figura 1. Grafo relacional

En la Tabla 2 presentamos el valor que toma cada una de las métricas para el ejemplo presentado.

Tabla 2. Valores de las métricas para el ejemplo

	NA(T)	NFK(T)	DRT(T)	RFK(T)	COS	NR	NA	NFK	DRT	RFK
Empleado	10	2	3	1/5						
Departamento	4	1	3	1/4						
Lugares_Depts	2	1	4	1/2						
Proyecto	4	1	4	1/4						
Trabaja_En	3	2	5	2/3						
Dependiente	5	1	4	1/5						
Esquema					36	1	28	8	5	2/7

3. BASES DE DATOS ACTIVAS

Las bases de datos activas son capaces de reaccionar ante un determinado evento. Estos sistemas proporcionan formas para describir, por un lado los eventos y sus reacciones asociadas (modelo de conocimiento) y por otro lado una estrategia que en tiempo de ejecución permite procesar y combinar este comportamiento activo (modelo de ejecución) (Paton y Díaz, 1999).

Una aproximación común al modelo de conocimiento es la utilización de reglas evento-condición-acción (ECA) en las cuales: el evento describe un acontecimiento ante el cual la regla es capaz de responder, la condición se encarga de examinar el contexto en el que el evento ha tenido lugar y la acción describe la tarea que la regla realizará si el evento tiene lugar y la condición se cumple. La parte del evento puede ser un evento primitivo (una operación en la base de datos), o un evento compuesto (combinación de eventos primitivos o eventos compuestos mediante operaciones del álgebra relacional).

El modelo de ejecución especifica como se tratan los conjuntos de reglas en tiempo de ejecución. Entre otras cosas, este modelo incluye el modelo de acoplamiento para determinar cuando la condición (acción) se evalúa (ejecuta) respecto al disparador (evaluación) del evento (condición).

De forma análoga al caso de las métricas para relacional, en este punto deberíamos plantearnos las preguntas que nos lleven a la definición de las métricas con el objetivo asegurar su mantenibilidad desde el punto de vista del diseñador de la base de datos

Para plantear estas preguntas, hemos de notar que la complejidad de la regla puede ser considerada como complejidad intrarregla cuando la regla sea medida de forma aislada e interregla cuando se está midiendo la interacción entre reglas.

Teniendo esto en cuenta, podemos intentar responder a la primera pregunta. En la interacción entre reglas, hay dos factores que influyen de forma más significativa:

- El ancho de la argumentación, el cual da una idea intuitiva sobre la longitud del flujo de eventos que conforma las circunstancias de las reglas. Lógicamente las reglas que se disparan por eventos compuestos son potencialmente más complicadas que las disparadas por un evento simple. Sin

embargo, lo que suele incrementar la dificultad es el hecho de que un mismo tipo de evento, que participa en el evento de una regla, se pueda producir en diferentes sitios.

- La profundidad de la argumentación, o sea, cómo de intrincada es la línea de razonamiento que conecta a la regla con el contexto en el que se han producido sus circunstancias.

Para medir las circunstancias de las reglas se puede recurrir a la utilización de un grafo de disparadores como el definido por Aiken et al. (1995). Un grafo de disparadores es un par $\langle S, L \rangle$ donde S representa el conjunto de reglas ECA y L es el conjunto de arcos dirigidos, donde un arco va de S_i a S_j si las acciones de S_i causan la ocurrencia de un evento que participa en los eventos de S_j .

Una variación de este grafo, propuesta por Díaz y Piattini (1999) consiste en, por un lado, ponderar los arcos con el número de eventos ocurrencia potenciales que pueden producirse por el disparo de una regla y, por el otro, ampliar los nodos con las transacciones (una transacción es un conjunto de acciones de la base de datos tal que esas acciones pueden corresponder a un evento que dispara una o más reglas).

En la Figura 2 se muestra un ejemplo de grafo extendido de disparadores, tomado de Díaz y Piattini (1999) donde $T0$ puede producir tres eventos significativos para $S4$ y uno para $S1$. La figura también muestra que los eventos de $S4$ pueden venir de $T0$, $S2$ y $S3$ los cuales potencialmente producen 3, 1 y 2 eventos significativos respectivamente.

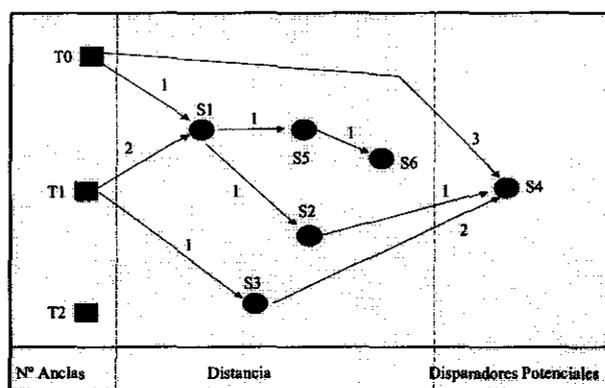


Figura 2. Ejemplo de grafo extendido de disparadores

Basándonos en los dos conceptos anteriores (ancho y profundidad de la argumentación), proponemos tres métricas interregla, en las que se considera que la cardinalidad del evento corresponde al número de eventos primitiva referenciados en la parte del evento de la regla independientemente del tipo de operador que se utilice para relacionarlos. Igualmente, sólo se considera el modo de acoplamiento inmediato en el que la condición (acción) es evaluada (ejecutada) inmediatamente después de que se produzca el evento (condición). Además, el ciclo de vida considerado es el ciclo de vida recursivo en el que los eventos que se producen durante la ejecución de la acción se tienen en cuenta inmediatamente, lo cual supone la suspensión de la acción actual continuando con ella en cuanto sea posible.

Estas tres métricas son:

Número de Anclas (NAn, *Number of Anchors*)

Definida como el número mínimo de anclas necesarias para abarcar el conjunto completo de causas potenciales de S_i . Un ancla es un nodo transacción del grafo que tiene un enlace (directo o transitivo)

con, al menos, una causa de S_i .

Distancia (D , *Distance*)

Esta métrica corresponde a la longitud del camino más largo que conecta S_i con cualquiera de sus anclas. Esta métrica mide la dificultad de la línea de razonamiento como el número de inferencias necesarias para conectar la causa última con su efecto.

Potencial de Disparo (TP, *Triggering potential*)

Dado un grafo de disparadores $\langle S, L \rangle$, y un nodo del grafo S_i , el número de causas de S_i es la suma de los pesos de los arcos que llegan a S_i . El valor de esta métrica para una regla será el cociente entre el número de causas potenciales de S_i y la cardinalidad de los eventos de S_i .

La Figura 3 muestra los valores de las métricas presentadas para distintos grafos de disparadores en los que la cardinalidad de los eventos de las reglas es uno y el peso de los arcos también es uno.

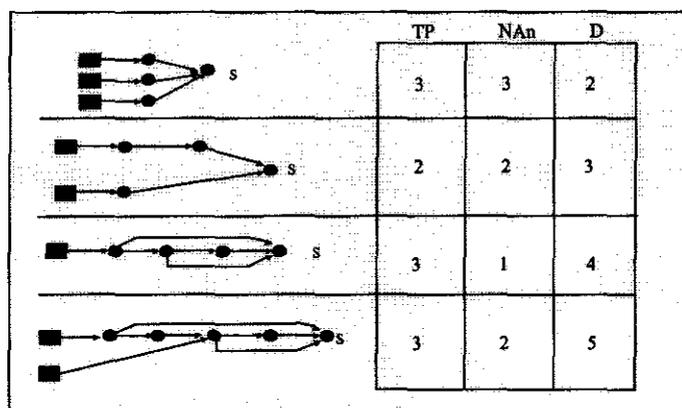


Figura 3. Valores de las métricas para tres ejemplos

Para responder a la segunda pregunta proponemos las siguientes métricas:

Número de Operandos (NO, *Number of Operands*).

Definida como el número de operandos que componen la condición

Número de Sentencias (NS, *Number of sentences*).

Definida como el número de sentencias SQL incluidas en la acción. Se excluye aquí la posibilidad de que la acción sea un programa PL/SQL ya que en ese caso podría directamente aplicarse cualquier métrica tradicional como la complejidad ciclomática propuesta por McCabe (1976).

Estas métricas cubren dos de las tres partes que componen una regla ECA: la condición (mediante la métrica número de operandos) y la acción (mediante la métrica número de sentencias). No consideramos la posibilidad de definir métricas para la parte del evento ya que la mayoría de los productos comerciales se limitan a soportar eventos simples.

Ejemplo

En la Figura 4, presentamos un disparador SQL y el valor de las métricas para el mismo.

```

CREATE TRIGGER CASO_CUATRO
AFTER DELETE ON P
FOR EACH ROW
WHEN (OLD.CIUDAD='PARIS')
BEGIN

```

NO= 1
 NS=3

Figura 4. Ejemplo de disparador y valor de las métricas

4. BASES DE DATOS OBJETO-RELACIONALES

Las bases de datos objeto-relacionales combinan las características de las bases de datos tradicionales (modelo de datos, seguridad, concurrencia, lenguaje de alto nivel...) con los principios de la orientación a objetos (encapsulamiento, generalización, agregación, polimorfismo...) (Stonebraker y Brown, 1999). Estos productos ofrecen la posibilidad de definir clases o tipos abstractos de datos además de las tablas de las bases de datos relacionales.

También, se pueden definir jerarquías de generalización entre clases (super y subclases) y entre tablas.

Los atributos de una tabla se pueden definir en un dominio simple (por ejemplo char) o en una clase (por ejemplo un número complejo, una imagen...). Estas clases pueden, además, formar parte de una generalización (Figura 5).

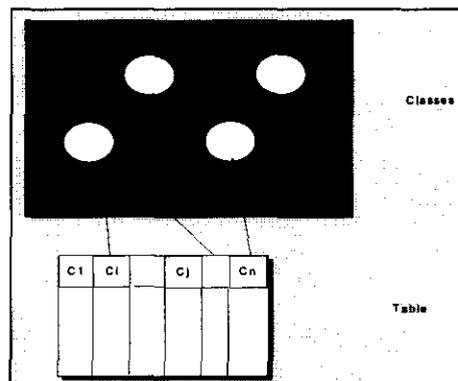


Figura 5. Ejemplo de definición de columnas complejas

Actualmente no existe un modelo para bases de datos objeto-relacionales comúnmente aceptado aunque estos conceptos básicos que hemos considerado se pueden encontrar en casi todas las propuestas.

Por otra parte, aunque el nuevo estándar SQL:1999 se ha aprobado recientemente (Eisenberg y Melton, 1999) todavía no se encuentra demasiado extendido en los productos existentes. Las consideraciones hechas en esta tesis son, en general, válidas para este estándar, aunque deberían ser

complementadas con otras métricas adicionales que capturen las características específicas de este estándar.

De nuevo debemos plantear las preguntas, en base a las características de estas bases de datos que nos permitan alcanzar el objetivo de asegurar su mantenibilidad desde el punto de vista del diseñador de la base de datos.

En este caso, las dos preguntas estarán de nuevo relacionadas con la complejidad de una tabla por un lado y con la complejidad de la interacción entre tablas por el otro, igual que ocurría para el caso de las bases de datos relacionales. La diferencia estriba en la complejidad de una tabla ya que, en este caso, la complejidad podrá ser debida a las columnas simples (aquellas definidas de un tipo de dato clásico, como INTEGER, DATE...) de la tabla o a las complejas (cada una de estas columnas complejas podrá ser una clase).

Para contestar a la primera pregunta hemos definido las siguientes métricas

- **Tamaño de la Tabla (TS, Table Size)**

Veremos que para definir esta métrica debemos definir otras que permiten contestar las otras preguntas relacionadas.

Definimos el tamaño de una tabla como el tamaño de sus columnas simples (*TSSC*) más el tamaño de sus columnas complejas (*TSCC*):

$$TS_i = TSSC + TSCC \quad (8)$$

- **Tamaño de las Columnas Simples (TSSC, Table Size Simple Columns)**

Esta métrica permite contestar a la pregunta 1.1. Considerando que todas las columnas simples tienen un tamaño igual a 1, el valor de *TSSC* corresponderá exactamente al número de atributos simples de la tabla (*NSA*):

$$TSSC = NSA \quad (9)$$

- **Tamaño de las Columnas Complejas (TSCC, Table Size Complex Columns)**

Esta métrica contesta a la pregunta 1.2. Definimos *TSCC* como la suma del tamaño de cada columna compleja (*CCS*):

$$TSCC = \sum_{i=1}^{NCC} CCS_i \quad (10)$$

siendo *NCC* el número de columnas complejas de la tabla

- **Tamaño de una Columna Compleja (CCS, Complex Column Size)**

El valor de *CCS* se obtiene con:

$$CCS = \frac{SHC}{NCU} \quad (11)$$

Siendo *SHC* el tamaño de la jerarquía a la que pertenece la clase sobre la que se define la columna y *NCU* el número de clases definidas sobre esa jerarquía. Si el número de columnas definidas sobre una clase es mayor que uno, la complejidad de la clase disminuye respecto a cada columna aunque no respecto al total de las columnas.

- **Tamaño de la Jerarquía sobre la que se Define una Columna Compleja (SHC, *Size Hierarchy Class*)**

SHC puede ser definido como la suma del tamaño de cada clase de la jerarquía (SC):

$$SHC = \sum_{i=1}^{NCH} SC_i \quad (12)$$

siendo *NCH* el número de clases en la jerarquía

- **Tamaño de una Clase (SC, *Size Class*)**

Y el tamaño de una clase se define como:

$$SC = \frac{SAC + SMC}{NHC} \quad (13)$$

siendo *SAC* la suma de los tamaños de los atributos de la clase, *SMC* la suma de los tamaños de los métodos de la clase y *NHC* el número de jerarquías a las que la clase pertenece.

- **Tamaño de los Atributos de una Clase (SAC, *Size Attributes of a Class*)**

Los atributos de una clase pueden igualmente ser simples o complejos por tanto *SAC* se define como la suma de los tamaños de los atributos simples (*SAS*, que tienen tamaño uno y por lo tanto se corresponde con el número de atributos simples) mas la suma de los tamaños de los atributos complejos (*CAS*) de la clase.

$$SAC = SAS + CAS \quad (14)$$

- **Tamaño de los Métodos de una Clase (SMC, *Size Methods of a Class*)**

Puede ser calculada mediante la versión de la complejidad ciclométrica de McCabe (1976) dada por Li y Henry (1993) ya presentada.

Y también podemos utilizar estas otras métricas:

- **Porcentaje de Columnas Complejas de una Tabla (PCC(T), *Percentage of complex columns of a table*).**

Definida como el porcentaje de columnas de la tabla *T* que son complejas, es decir definidas sobre una clase.

$$PCC = \frac{NCC(T)}{NC(T)} \quad (15)$$

donde *NC(T)* es el número de columnas que tiene la tabla *T* y *NCC(T)* es el número de columnas complejas de la tabla *T*.

- **Número de Clases Involucradas en la Definición de una Tabla (NIC(T), *Number of involved classes of a table*)**

Definida como el número de clases que componen los tipos de las columnas complejas de la tabla *T* mediante generalización o agregación.

- **Número de Clases Compartidas (NSC(T), *Number of shared classes of a table*).**

Definida como el número de clases que se usan para la definición de alguna columna de la tabla *T* y que también son utilizadas por otras tablas.

Igualmente, debido al soporte relacional de este tipo de bases de datos también podremos utilizar gran parte de las métricas definidas para bases de datos relacionales, a saber: **Número de Claves Ajenas de una Tabla (NFK)**, **Profundidad del Árbol Referencial de una Tabla (DRT(T))**, **Ratio de Claves Ajenas de una Tabla (RFK(T))**.

Para contestar a la segunda pregunta definimos las siguientes métricas específicas para bases de datos objeto-relacionales:

- **Tamaño del Esquema (SS, *Schema Size*).**

Definida como la suma del tamaño de las tablas que componen el esquema. La expresión matemática para esta métrica es:

$$SS = \sum_{i=1}^{NT} TS_i \quad (16)$$

Siendo NT el número de tablas del esquema y TS la métrica “tamaño de una tabla” presentada en la sección anterior.

También podemos utilizar algunas de las métricas presentadas anteriormente, pero orientadas a esquema:

- **Porcentaje de Columnas Complejas (PCC, *Percentage of complex columns*).**

Definida como el porcentaje de columnas de todas las tablas del esquema que son complejas, es decir definidas sobre una clase.

$$PCC = \frac{NCC}{NC} \quad (17)$$

donde NCC es el número de columnas complejas que hay en las tablas del esquema

$$(NCC = \sum_{i=1}^{NT} NCC(T_i)) \quad (18)$$

y NC es el número de columnas totales que hay en las tablas del esquema

$$(NC = \sum_{i=1}^{NT} NC(T_i)) \quad (19)$$

- **Número de Clases Involucradas en la Definición (NIC, *Number of involved classes*)**

Definida como el número de clases que componen los tipos de las columnas complejas de todas las tablas del esquema mediante generalización o agregación.

- **Número de Clases Compartidas (NSC, *Number of shared classes*).**

Definida como el número de clases compartidas por las tablas del esquema.

Igualmente, debido a las características relacionales de este tipo de bases de datos también podremos utilizar todas las métricas definidas para bases de datos relacionales, a saber: **Cohesión del Esquema (COS)**, **Número de Tablas (NT)**, **Número de Claves Ajenas (NFK)**, **Profundidad del Árbol Referencial (DRT)**, **Ratio de Claves Ajenas (RFK)**.

Ejemplo

A continuación proponemos un ejemplo y el valor que las métricas presentadas toman para el mismo (Tabla 3).

Únicamente se presentarán las métricas específicas para modelos objeto-relacionales.

Tabla 3. Ejemplo de tabla con soporte objeto-relacional

<pre>CREATE TABLE casa(idcasa INTEGER, idagencia INTEGER, precio INTEGER, habitaciones INTEGER, tamaño DECIMAL (8,2), situación dirección, desc text, vista_frontal bitmap, documento doc, vendedor empleado, PRIMARY KEY (idcasa), FOREIGN KEY idagencia REFERENCES agencia(id)); CREATE TABLE agencia(id INTEGER, nombre VARCHAR(20), localización dirección);</pre>	<pre>CREATE TYPE dirección AS(calle CHAR(30), pueblo CHAR(20), ciudad CHAR(2), cod_postal INTEGER) NOT FINAL; CREATE TYPE empleado AS(nombre CHAR(40), salario_base DECIMAL(9,2), bonos DECIMAL(9,2)) INSTANTIABLE NOT FINAL METHOD salario() RETURNS DECIMAL(9,2); CREATE METHOD salario() FOR empleado BEGIN ... END;</pre>
---	---

Para simplificar los cálculos vamos a considerar que tanto los métodos como los objetos largos (LOBs, como texto o bitmap) tienen también un tamaño igual a uno.

Calculamos primero los valores para las clases dirección y empleado:

$$CS_{dirección} = 4$$

$$CS_{empleado} = 3 + 1 = 4$$

$$CCS_{situación} = \frac{4}{2} = 2$$

$$CCS_{localización} = \frac{4}{2} = 2$$

$$CCS_{vendedor} = \frac{4}{1} = 4$$

Y con estos valores podemos obtener los tamaños de cada columna de cada tabla. Estos valores se presentan en la Tabla 4.

Tabla 4. Tamaño de cada columna

	Nombre Columna	Tipo Columna	Tamaño Columna
Casa	Idcasa	Simple	1
	Idagencis	Simple	1
	Precio	Simple	1
	Habitaciones	Simple	1
	Tamaño	Simple	1
	Situación	Complejo	2
	Desc	LOB	1
	Vista frontal	LOB	1
	Documento	LOB	1
	Vendedor	Complejo	4
Agencia	Id	Simple	1
	Nombre	Simple	1
	Localización	Complejo	2

Con estos datos, obtenemos los siguientes valores para el tamaño de cada una de las tablas:

$$TS_{casa} = 5 + 9 = 14$$

$$TS_{agencia} = 2 + 2 = 4$$

Y por lo tanto, a partir de estos valores, podemos obtener el valor de la métrica SS (Tamaño del esquema):

$$SS = TS_{casa} + TS_{agencia} = 14 + 4 = 18$$

El resto de las métricas para las tablas casa y agencia se presentan en la Tabla 5.

Tabla 5. Valores de las métricas para el ejemplo de la tabla 3

	Casa	Comentarios	Agencia	Comentarios
TS	14	Ya vista	4	Ya vista
NFK	1	idagencia	0	No tiene claves ajenas
DRT	1	Casa a Agencia	0	No tiene claves ajenas
PCC	20%	2 complejos sobre 10	33%	1 complejo sobre 3
NIC	2	Dirección y empleado	1	Dirección
NSC	1	Dirección	1	Dirección

5. VALIDACIÓN DE LAS MÉTRICAS

En esta sección presentaremos un resumen de las validaciones formales y empíricas realizadas con las métricas expuestas en las secciones anteriores.

5.1. Validación Formal

La validación formal de las métricas ha sido hecha siguiendo las dos tendencias principales (las aproximaciones axiomáticas, con el marco de Briand et al (1996) y la teoría de la medida, con el marco de Zuse (1998)). En la Tabla 6, presentamos los resultados obtenidos.

Tabla 6. Sumario de la validación formal de las métricas propuestas

Base De Datos	Nivel	MÉTRICA	Briand Et Al. (1996)	Zuse (1998)
Relacional	Tabla	NA	Tamaño	Encima Ordinal
		NFK	Complejidad	Encima Ordinal
		DRT	Longitud	Ordinal
		RFK	No Clasificable	Absoluta
	Esquema	NA	Tamaño	Encima Ordinal
		NFK	Complejidad	Encima Ordinal
		DRT	Longitud	Ordinal
		RFK	No Clasificable	Absoluta
		NT	Tamaño	Ratio
		NR	No Clasificable	Absoluta
Activas	Intra	CO	No Clasificable	Ratio
		NAn	Complejidad	Ordinal
		TP	No Clasificable	Ordinal
	Inter	D	Longitud	Ordinal
		NO	Tamaño	Ratio
Objeto-Relacional	Tabla	NS	Tamaño	Ratio
		TS	Tamaño	Encima Ordinal
		TSSC	Tamaño	Encima Ordinal
		TSCC	Tamaño	Encima Ordinal
		CCS	Tamaño	Ratio
		SHC	Tamaño	Encima Ordinal
		SC	Tamaño	Encima Ordinal
		SAC	Tamaño	Encima Ordinal
		SMC	Tamaño	Intervalo
		PCC	No Clasificable	Absoluta
		NIC	Tamaño	Ratio
		NSC	Tamaño	Encima Ordinal
		Esquema	NFK	Complejidad
	DRT		Longitud	Ordinal
	RFK		No Clasificable	Absoluta
	SS		Tamaño	Ratio
	NIC		Tamaño	Ratio
	PCC		No Clasificable	Absoluta
	NSC		Tamaño	Encima Ordinal
	CO	No Clasificable	Ratio	
NT	Tamaño	Ratio		
NFK	Complejidad	Encima Ordinal		
DRT	Longitud	Ordinal		
RFK	No Clasificable	Absoluta		

El desarrollo completo de estas validaciones puede ser encontrado en las referencias que se muestran en la Tabla 7.

Tabla 7. Referencias para la validación formal

	Briand Et Al (1996)	Zuse (1998)
Relacionales	Piattini, M., Calero, C., Polo, M. y Ruiz, F. (2000).	Piattini, M., Calero, C y Genero, M (2001)
Activas	Calero (2001)	Díaz, O., Piattini, M. y Calero, C. (2001)
Objeto-Relacionales	Calero (2001)	Piattini, M., Calero, C., Sahraoui, H y Lounis, H. (2001)

5.2. Validación Empírica

La validación empírica de las métricas se ha hecho mediante diferentes experimentos controlados y casos de estudio. En la Tabla 8 presentamos un resumen del trabajo llevado a cabo.

En la Tabla 9 se pueden encontrar las referencias dónde se encuentran estos estudios empíricos detallados.

Tabla 8. Sumario de la validación empírica de las métricas propuestas

	Experimento Controlado		Réplica		Caso De Estudio	
	Métricas	Drt, Rfk	Métricas	Drt, Rfk	Métricas	Nt, Na, Rfk
Relacional	Sujetos	Alumnos	Sujetos	Expertos	Sujetos	Expertos
	Métricas	Tp, D	Métricas	Tp, D		
Activas Inter.	Sujetos	Alumnos	Sujetos	Expertos		
	Métricas	Ns, No	Métricas	Ns, No		
Activas Intra	Sujetos	Alumnos	Sujetos	Alumnos		
	Métricas	Pcc, Ts, Drt, Rfk, Nic, Nsc	Métricas	Pcc, Ts, Drt, Rfk, Nic, Nsc		
Objeto-Relacional	Sujetos	Expertos	Sujetos	Expertos		

Tabla 9. Referencias para la validación empírica

	Experimento	Caso de Estudio
Relacionales	Calero, C, Piattini, M. y Genero, M (2001a)	Calero, C, Piattini, M. y Genero, M (2001b)
Activas	Díaz, O., Piattini, M. Y Calero, C. (2001)	
Objeto-Relacionales	Calero, C., Sahraoui, H., Piattini, M. y Lounis, H. (2001)	

6. CONCLUSIONES

En este artículo se han presentado distintas métricas para bases de datos, lo cual consideramos de vital importancia, debido no sólo al papel protagonista que las bases de datos tienen en los SI actuales sino también a la total inexistencia de métricas específicas para bases de datos. La definición de estas métricas se ha hecho teniendo en cuenta las características específicas de aquello que se quería medir, procurando también obtener métricas útiles, mediante la aplicación de la aproximación GQM en conjunción con diferentes normas ISO (principalmente la ISO 9126). Además, todas las métricas presentadas han sido validadas, tanto formal como empíricamente, habiéndose presentado los principales resultados.

REFERENCIAS

- [1] V.R. Basili y D. Weiss, "A Methodology for Collecting Valid Software Engineering Data". *IEEE Transactions on Software Engineering*, SE-10. No. 6, pp.728-738, 1984.
- [2] C. Calero, M. Piattini y M. Genero, "Empirical Validation of Referential Integrity Metrics". *Information and Software Technology. Special Issue on Controlled Experiments in Software Engineering*, Vol. 43, No.15, pp. 949-957, 2001.
- [3] C. Calero, M. Piattini y M. Genero, "A Case Study with Relational Database Metrics". ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2001). IEEE Computer Society. ISBN: 0-7695-1165-1, 2001.
- [4] C. Calero, *Definición de un Conjunto de Métricas para la Mantenibilidad de Bases de Datos Relacionales, Activas y Objeto-Relacionales*, tesis doctoral, Universidad de Castilla-La Mancha, 2001.
- [5] C. Calero, H. Sahraoui, M. Piattini y H. Lounis, "Estimating Object-Relational Database Understandability Using Structural Metrics", *12th International Conference on Database and Expert Systems Applications (DEXA 2001)*, Lecture Notes in Computer Science 2113, Springer-Verlag, ISBN: 3-540-42527-6, pp. 909-922, 2001.
- [6] D.N. Card y R.L. Glass, *Measuring Software Design Quality*, Englewood Cliffs, E.U.A., 1990.
- [7] C.J. Date, *An Introduction to Database Systems*, 6th Ed., Addison-Wesley Reading, Massachusetts, 1995.
- [8] A. De Miguel, M. Piattini y E. Marcos, *Fundamentos y Modelos de Bases de Datos*, 2^a Ed., Rama, Madrid, 1999.
- [9] O. Díaz y M. Piattini, "Metrics for Active Databases Maintainability", *CAISE'99*, Heidelberg, Lecture Notes in Computer Science, LCNS 1626, Springer-Verlag, M. Jarke y A. Oberweis (Eds.), pp. 472-476, ISBN 3-540-66157-3, 1999.
- [10] O. Díaz, M. Piattini y C. Calero, "Measuring Active Databases Maintainability", *Information Systems Journal*, Vol. 26, pp. 17-36, 2001.
- [11] A. Eisenberg y J. Melton, "SQL: 1999, Formerly Known as SQL3". *SIGMOD Record*, Vol., 28, No. 1, pp. 131-138, 1999.
- [12] R. Elmasri y S. Navathe, *Database Systems*, Second Ed., Addison-Wesley, Massachusetts, 1997.
- [13] B. Henderson-Sellers, *Object-oriented Metrics - Measures of Complexity*, Prentice-Hall, Upper Saddle River, New Jersey, 1996.
- [14] ISO 9126, *Software Product Evaluation-Quality Characteristics and Guidelines for their Use*, ISO/IEC Standard 9126, Geneva, 1999.
- [15] B. Kitchenham y S.J. Linkman, "Design Metrics in Practice", *Inf. Soft. Technology*, Vol. 32, No. 4, pp.304-310, 1990.
- [16] A.M. Kotz, "Active Database Functionality in a Real-World Banking Environment", *Dagstuhl Seminar on Active DBMS*, 1994.
- [17] N. Leavitt, "Whatever Happened to Object-Oriented Databases?", *IEEE Computer Society*, pp. 16-19, Agosto. 2000.
- [18] H.F. Li y W.K. Cheng, "An Empirical Study of Software Metrics", *IEEE Trans. on Software Engineering*, Vol. 13, No. 6, pp. 679-708, 1987.
- [19] S.G. MacDonell, M.J. Shepperd y P.J. Sallis, "Metrics for Database Systems: An Empirical Study", *Proc. Fourth International Software Metrics Symposium - Metrics'97*, Albuquerque, IEEE Computer Society, pp. 99-107, 1997.
- [20] T.J. McCabe, "A Complexity Measure", *IEEE Trans. Software Engineering*, Vol. 2, No.5, pp.308-320, 1976.
- [21] M. Paton y O. Díaz, "Active Database Management Systems", *ACM Computing Surveys*, Vol. 31, No. 1, pp.63-103, 1999.

- [22] M. Piattini, C. Calero, M. Polo y F. Ruiz, "Towards a Metric Suite for Relational Database Complexity", *Fifth World Conference on Integrated Design and Process Technology IDPT 2000*, Dallas (USA), 4-8 June 2000, CD ROM del congreso, ISSN.1090-9389, 2000.
- [23] M. Piattini, C. Calero y M. Genero, "Table Oriented Metrics for Relational Databases", *Software Quality Journal*, Vol. 9, pp. 79-97, 2001.
- [24] T.M. Pigoski, "*Practical Software Maintenance*", Wiley Computer Publishing, New York, E.U.A., 1997.
- [25] H.D. Rombach, "Design Measurement: Some Lessons Learned", *IEEE Software*, Vol. 7, No. 3, pp.17-25, 1990.
- [26] H.M. Sneed y O. Foshag, "Measuring Legacy Database Structures", *Proc of The European Software Measurement Conference, FESMA 98*, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (Eds.), pp.199-211, 1998.
- [27] M. Stonebraker y P. Brown, *Object-Relational DBMSs Tracking the Next Great Wave*, California, Morgan Kauffman Publishers, 1999.